February 1981
US$2.50/DM 9

80 microcomputing

the magazine for TRS-80* users

Education . . .

A profound change?

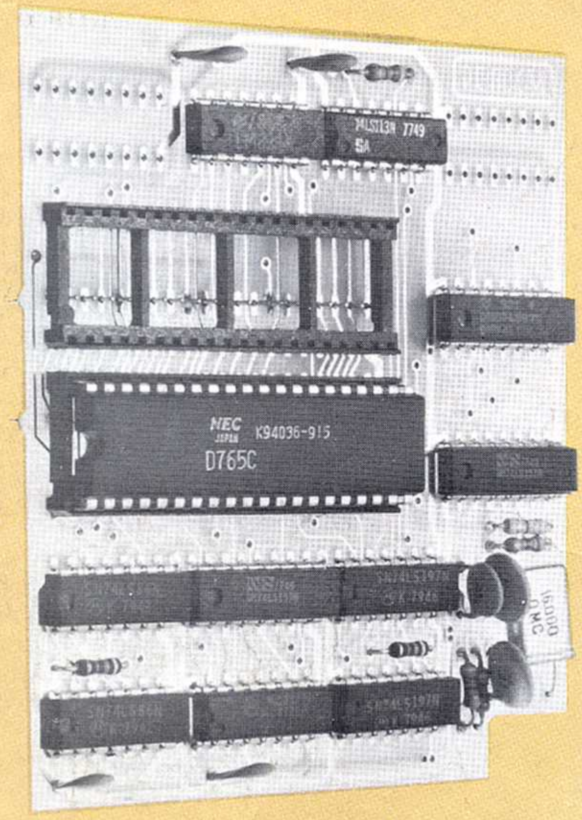
Special Reports:
Micros in the classroom:
Cumberland, Indiana
Rosemount, Minnesota
Durham, North Carolina
Westwood, Massachusetts

Plus: an expanded review section of educational software and books.

*trademark of Tandy Corp.

TRS-80* Model I Computer Owners . . .

Store Up to 350 Kbytes on a 5″ Disk



The DOUBLER™. It packs almost twice the data on a disk track as your single-density system. Depending on the type of drive, you can store up to **four** times more data on one side of a minidiskette than you can store using a standard Model I mini-disk drive.

- The DOUBLER™ reads, writes and formats **either** single- or double-density minidiskettes.
- Proprietary design allows you to continue to run TRSDOS*, NEWDOS‡, Percom OS-80™ or other single-density software **without making any changes** to software or hardware. Switch to double-density operation at any convenient time.
- Includes DBLDOS™, a **TRSDOS* compatible** double-density disk operating system.
- CONVERT utility, on DBLDOS™ minidiskette, converts files and programs from single- to double-density or double- to single-density.
- The DOUBLER™ circuit card **includes high performance data separator, write precompensation** circuits for reliable disk read operations — even with 80-track drives.
- **Plug-in Installation** — The DOUBLER simply plugs into the disk controller socket of your Expansion Interface, requiring no strapping or trace cutting. Expansion Interface disk controller may be completely restored to original configuration by simply removing the DOUBLER™ and re-installing the original disk controller chip.
- Works with standard 35-, 40-, 77- and 80-track mini-disk drives rated for double-density operation.
- Introductory price, including DBLDOS™ and format conversion utility on minidiskette, **only** $219.95.

Mini-Disk Systems

More storage capacity, higher reliability — from Percom, the industry leader. One-, two- and three-drive configurations in either 40- or 77-track format. Fully burned-in. From only $399.

Double-Density Software

(On diskette — with instruction manual.)

OS-80D™ Double-Density Disk Operating System — This double-density upgrade version of Percom's acclaimed OS-80™ resides *entirely* in RAM — requiring only 7.5-Kbytes! A BASIC programmer's "dream operating system," even utilities are in BASIC.

DOUBLEZAP-II/80 This program modifies Apparat's NEWDOS/80‡ to run either double — or single-density programs — even to run a mix of the two formats on one system!

DOUBLEZAP-II/V This program modifies Virtual Technology's VTOS 4.0†† to provide the same capability as DOUBLEZAP-II/80 provides for NEWDOS/80.

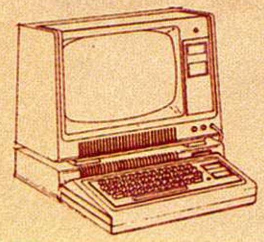
Call toll-free, 1-800-527-1592, for the address of your nearest authorized Percom dealer, or to order directly from Percom.

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

™ trademark of Percom Data Company, Inc.
* trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.
trademark of Apparat Company, Inc. ††trademark of Virtual Technology, Inc.

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.
PRICES DO NOT INCLUDE SHIPPING AND HANDLING CHARGES.

TRS-80* Model I Computer Owners . . .



Double-density storage. It's really here!

Here at Percom. And your authorized Percom dealers.

And double-density storage is here in a big way. Because now you can choose from *three different levels* of mini-disk systems — all *double-density rated.*

And get the storage that precisely meets your application needs.

Not to mention the service and quality that's made Percom the industry leader.

Although rated for double-density operation, all levels of Percom drives *work equally well in single-density* applications.

You can operate these drives in ordinary single-density format using TRSDOS*, Percom OS-80™ or any other single-density operating system.

Or, you can add a Percom DOUBLER™ to your Tandy Expansion Interface and store data and programs in *either* single- or double-density format.

Under double-density operation, you can store as much as *350 Kbytes* of formatted data — depending on the drive model — on one side of a five-inch minidiskette. That's *four times* the capacity of standard 35-track Model I minidisks, almost 100 Kbytes *more than* the capacity of the *eight-inch* IBM 3740 format!

Available in 1-, 2- and 3-drive configurations in all three model lines, Percom *burned-in, fully-tested* drives start at only $399.

TFD-40™ Drives

TFD-40 Drives store 180 Kbytes (double-density) or 102 Kbytes (single-density) of **formatted** data on one side of a 40-track minidiskette. Although economically priced, TFD-40 drives receive the same full Percom quality control measures as TFD-100 and TFD-200 drives.

TFD-100™ Drives

TFD-100 drives are "flippy" drives. You store twice the data per minidiskette by using both sides of the disk. TFD-100 drives store 180 Kbytes (double-density) or 102 Kbytes (single-density) **per side.** Under double-density operation, you can store a 70-page document on one minidiskette.

TFD-200™ Drives

TFD-200 drives store 350 Kbytes (double-density) or 197 Kbytes (single-density) on one side of a minidiskette. By comparison, 3740-formatted eight-inch disks store only 256 Kbytes. Enormous on-line storage capacity in a 5" drive, plus proven Percom reliability. That's what you get in a TFD-200.

The DOUBLER™ — This proprietary adapter for the TRS-80* Model I computer packs approximately twice the data on a disk track.

Depending on the type of drive, you can store up to four times as much data — 350 Kbytes — on one side of a minidiskette as you can store using a Tandy standard Model I computer drive.

Easy to install, the DOUBLER merely plugs into the disk controller chip socket of your Expansion Interface. No rewiring. No trace cutting.

And because the DOUBLER reads, writes and formats *either* single- or double-density disks, you can continue to run all of your single-density software, then switch to double-density operation at any convenient time.

Included with the PC card adapter is a TRSDOS*-compatible double-density disk operating system, called DBLDOS™, plus a CONVERT utility that converts files and programs from single- to double-density or double- to single-density format.

Each DOUBLER also includes an on-card high-performance *data separator circuit* which ensures reliable disk read operation.

The DOUBLER works with standard 35-, 40-, 77- and 80-track drives rated for double-density operation.

Note. Opening the Expansion Interface to install the DOUBLER may void Tandy's limited 90-day warranty.

Free software patch This software patch, called PATCH PAK™, upgrades TRSDOS* for operation with improved 40- and 77-track drives. For single-density operation only.

Quality Percom products are available at authorized dealers. Call *toll free 1-800-527-1592* for the address of your nearest dealer or to order directly from Percom. In Canada call 519-824-7041.

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY • GARLAND TX • 75042
(214) 272-3421

™ trademark of Percom Data Company, Inc.
mark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.

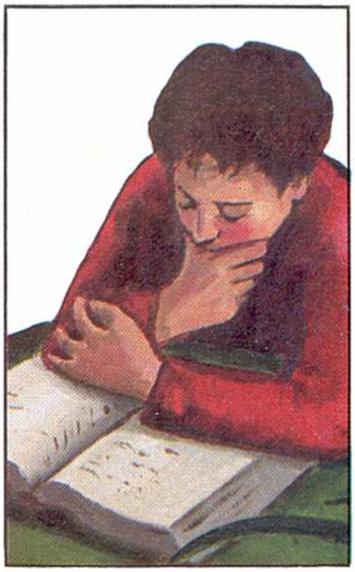
Prices and specifications subject to change without notice.

✓408

80 Contents

PUBLISHER
Wayne Green
EXECUTIVE VICE PRESIDENT
Sherry Smythe
CORPORATE CONTROLLER
Charles Garniss, Jr.
ASSOCIATE PUBLISHER
Edward Ferman
ASSISTANT PUBLISHER
Jeff DeTray
ADVERTISING MANAGER
Kevin Rushalko
CIRCULATION MANAGER
(603) 924-7296
Debra Boudrieau
BULK SALES MANAGER
Ginny Boudrieau
ADVERTISING SALES
(603) 924-7138
Penny Brooks
John Gancarz

80 Microcomputing (ISSN ·0199-6789) is published monthly by 1001001 Inc., 80 Pine St., Peterborough NH 03458. Phone: 603-924-3873. Subscription rates in U.S. are $18 for one year and $45 for three years. In Canada, $20—one year only, U.S. funds. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. Foreign subscriptions (surface mail), $28—one year only, U.S. funds. Foreign subscriptions (air mail), $60—one year only, U.S. funds. In Europe contact Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. In South Africa contact *80 Microcomputing*, P.O. Box 782815, Sandton, South Africa 2146. Australian Distributor: Electronic Concepts, Attention: Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. All U.S. subscription correspondence should be addressed to *80 Microcomputing*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form ·3579 to *80 Microcomputing*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737.



COMING NEXT MONTH

80 will take an overview of the popular Disk Operating Systems. Stewart Fason and John Burgan take a close look at NEWDOS+, NEWDOS-80, UltraDOS, DOSPLUS, and VTOS 4.01. What are the distinctions? How do they compare in value? What's the documentation like?

For those of you still wired by cassette, Jake Commander will take you on a guided tour of the Disk Operating System and how it works.

PUBLISHER/EDITOR
Wayne Green

MANAGING EDITOR
Michael Comendul

TECHNICAL CONSULTANT
Jake Commander

PRODUCTION EDITOR
Clare McCarthy

NEWS EDITOR
Nancy Robertson

REVIEW EDITOR
Pamela Petrakos

ASST. TECHNICAL EDITOR
Chris Brown

ASSISTANT EDITORS
Chris Crocker
Debra Marshall

TECHNICAL CONTRIBUTING EDITOR
Dennis Kitsz

EDITORIAL ADMINISTRATION
Cresca Clyne
Pat Graham
Nancy Noyd

DESIGN ASSOCIATE
Diana Shonk

MANUFACTURING MANAGER
Noel Ray Self

PRODUCTION MANAGER
Nancy Salmon

ASST. PRODUCTION MANAGER
Michael Murphy

AD GRAPHICS MANAGER
Robert Drew

AD COORDINATOR
Sue Symonds

ADVERTISING PRODUCTION
Steve Baldwin, Bruce Hedin, Maryann Metivier, Dion Owens

PRODUCTION DEPT.
Joan Ahern, William Anderson Jr., Linda Drew, Bob Dukette, Kenneth Jackson, Ross Kenyon, Theresa Ostebo, Jane Preston, Patrice Scribner, Thomas Villeneuve

PHOTOGRAPHY
William Heydolph, Terrie Anderson, Bill Suttenfield

TYPESETTING
Barbara Latti, Sara Bedell, Michele Desrocher, Luann Keddy, Mary Kinzel, Linda Locke, Karen Podzycki

Cover illustration by Alison Hanft

80 REMARKS

by Wayne Green

"Until we know enough about the mind to understand how to access at will material recorded in the unconscious, we should consider the impact of this 'use it or lose it' concept."

Computers will, I hope, bring about a major change in education, second only to the printing press.

The educational system I envision for our future is much more than the "computer aided education" (CAI) which has developed so far. Try not to project your impressions of programmed education and computerized drills into the world of the year 2000. Instead, let's consider the strong points of our electronic technical developments and contrast them with the weak points of the present educational system.

Much of the educational process is boring for many people and this tends to limit the absorption of material. I remember, all too well, falling asleep in response to my efforts to memorize French vocabulary in high school.

Perhaps, if we think in terms of helping people to learn rather than "teaching" them it will help us focus on the future. There is a fundamental difference and it involves personal freedom—learning because we *want* to, not because we are being *forced* to. I hope that the computer may free people from the twelve to twenty years of virtual slavery they now endure, allowing them to learn those things of interest and value to them when they want to learn them and at the speed best suited to them.

An enormous percentage of the things we are "taught" are forgotten and the effort wasted. Until we know enough about the mind to understand how to access at will material recorded in the unconscious, we should consider the impact of this "use it or lose it" concept.

A few years ago I met a ham operator in Paris, Pierre Catala F2BO, and he wanted to come to the U.S. I helped him come over and he worked for me for a while—then went to college and went to work in communications. He returned to Paris to visit his mother after having been away perhaps three years. He found that in this short time he had lost his facility in French and that many words escaped him.

The electronic "school" that I envision for the future may at first frighten teachers. Panic not, and see me through this. Teachers will be needed, as much as ever, and may even be more appreciated in the future.

Video and the Computer

Some parts of education are well adapted to our recent developments in electronics—some aren't. Much of what is being taught in school might be better communicated to students via a combination television recorder and computer.

Suppose we were to produce programs like those in the "Nova" and "Connections" series on P.B.S. as a method of helping people understand most of the subjects currently taught in school? Every now and then the program would stop and the computer would ask questions to make sure the material was being understood. The system could either continue or go back automatically and replay the part not understood. This allows each person to progress at his own speed, thus solving one of the great problems facing teachers today—the difference in learning speed of students.

"Much of the educational process is boring. . . and this tends to limit. . . absorption. . . ."

With millions of people using each program, we need not stint on their production. Our cameras can travel anywhere and simulate any situation which will help us learn. The world's foremost authorities will be our tutors. Our teaching materials will essentially be unlimited.

With my "electronic school" any interest can be pursued. Perhaps I want to know about, say, wallpaper. I could watch a program on the history of wallpaper—then one on how it is made—followed by one on how to put it up. Then I might want to know about patterns of wallpaper which are available—or run another program on the use of wallpaper and interior decorating.

Project that concept to everything we see around us and you have an almost unlimited and very practical education—one which can continue for a lifetime.

Where does the teacher fit into all this? First, I think that students will want a teacher for extra help. Help could be accessed via a two-way television system, perhaps over cable—when people are working from home—or remote from a needed expert.

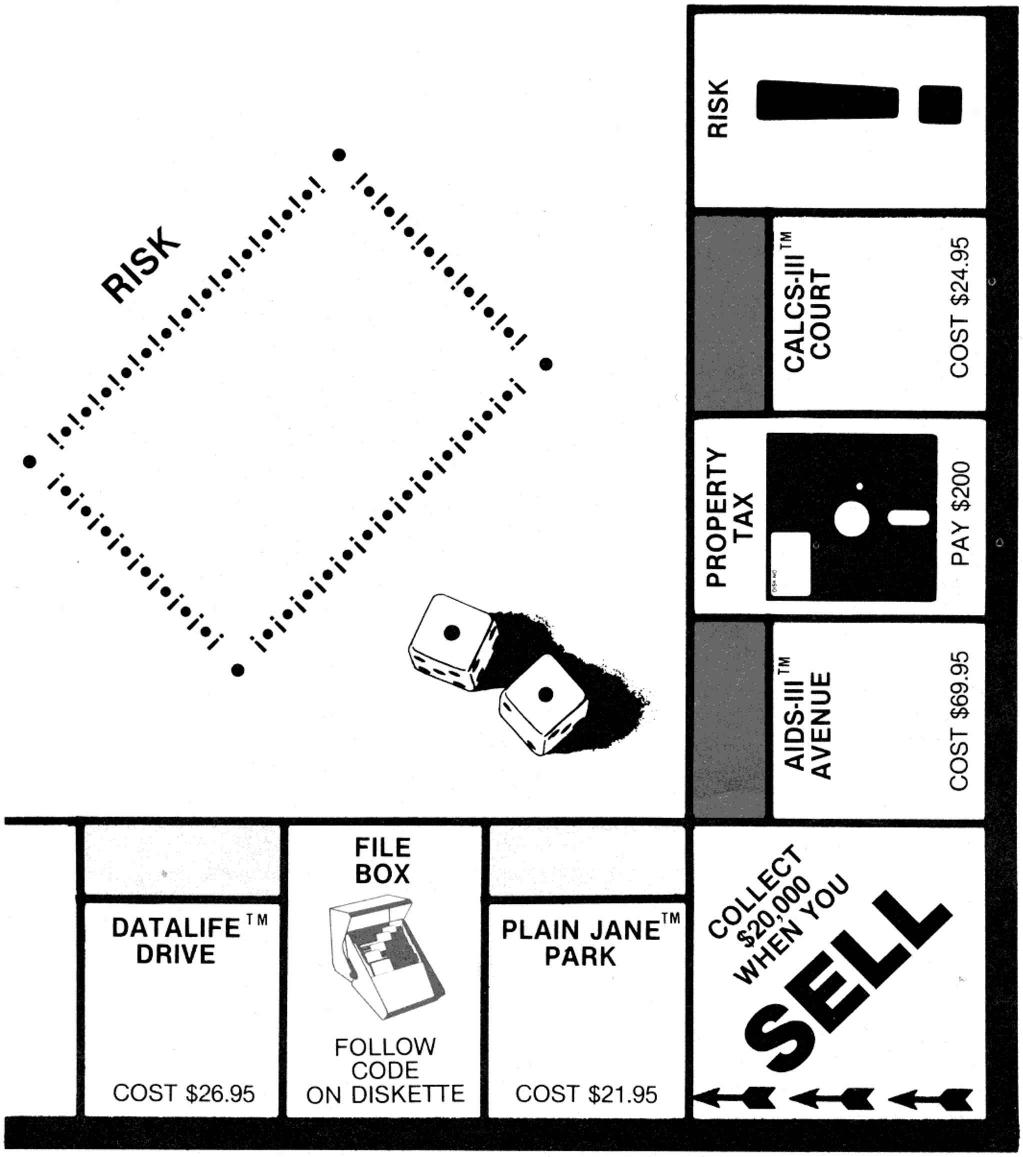
Second, I believe that there will be a great interest in acquiring skills—and we will need teachers and school facilities for this. You have to have metalworking machines to learn metalworking, so too, woodworking, electronic construction and testing and hundreds of other personal skills.

Not only will the student of the future be able to learn anything of interest through his life, but he will be able to learn at many times the present rate. And he'll enjoy it!

In the school building of the future we may be able to learn to swim, skin-dive, drive, fly, cook, sew, do chemical experiments, use scientific equipment, telescopes, learn musical instruments, casting, painting, sculpture, juggling, magic, hang gliding, ballooning, skiing, public speaking, plumbing, cement making, pottery making, lock repair, roller skating, ice skating, welding, boat handling, acting, selling, haircutting, dancing, movie making, photography, navigation, map reading, basket weaving (I had to put *that* in), gardening, and many other skills involving dexterity and coordination. Oh, we'll be needing teachers and we will need much better equipped schools than we have today.

Providing for the future of learning is going to take a lot of work, but the benefits for every one involved are outstanding. For my part, I shall be in there pushing for the changes which I think will make things better. This publication is a start. *80* will try to provide a forum for professional educators in this and future issues. As interest in the use of computers in education grows I plan to spin off a publication just for this field. It is hoped that articles on successful educational systems in this and future issues will bring commercial firms and educators together. ■

WHY TAKE A CHANCE WITH YOUR ADVERTISING?



YOU DON'T TREAT YOUR BUSINESS LIKE A GAME . . .

And neither do we. META MEDIA can help you reach your market more effectively and for less money than you may think you have to spend. Business is risky enough. Why not deal with people who understand products for small computers and how to sell them. You have a lot at stake . . . time, money, your future. Why roll the dice?

Meta Media Productions, Inc. ↙219

26111 Brush Avenue Euclid, Ohio 44132. (216) 289-1100

We Accept: VISA/MasterCard

DATALIFE is a TM of VERBATIM
AIDS-III, CALCS-III and PLAIN JANE are TM of MTC

META MEDIA PRODUCTIONS, INC. is an affiliate of META TECHNOLOGIES CORPORATION, INC.

INSIDE 80

by Ed Juge, director of
computer merchandising, Tandy Radio Shack

"Don't become the next victim of the Radio Shack 'Black Hole' where orders for unknown products end up!"

By now many of you will have purchased the Model III versions of Scripsit and Visicalc, and based on calls we began getting in November and December (before the packages were even released), I suspect that the mill is churning out even more rumors now. Both packages contain a "limited backup" protection system, a subject that seems to concern some prospective buyers. Let's put some of the rumors to rest.

The protection system limits the number of copies which can be made from an original "protected" program. Only that program is affected. You can backup the disk five or 500 times—there's no limit. Visicalc and Scripsit (being the protected programs on their diskettes) appear only on the first two backups. Those backups then can't be copied further.

Specific Uses

We plan to use this system only on specific types of programs—those written so you don't have to operate the program with the program disk resident in a drive. You'll put your program disk in Drive 0, load the program, then remove the program disk and tuck it safely away. You then insert a plain old TRSDOS disk in Drive 0 for data handling.

Rumors began back in November that we plan to protect TRSDOS. Not so! You can back TRSDOS up as often as you wish for your own use. Remember, however, that TRSDOS is a copyrighted product. If you want to sell your original software creation on a TRSDOS disk, you need to contact us for information on how to do it without violating the copyright laws. (Write to Mr. G. V. Pack, Staff Attorney, Tandy Corp., 1800 One Tandy Center, Ft. Worth, TX 76102 for details.)

So what happens if you *do* blow all three copies of a protected program? Well, be sure you save your original disk—the one with our label on it. Use it *second . . . after* you've blown one of the backups. Then, should you blow that disk, you have one more backup to operate on while you take the original to your Radio Shack store so they can mail it in for replacement. Since you only use these disks to load the program, three of them should last for years!

Speaking of Visicalc, there is an enhanced Model III version in the wings. Those of you who buy (or bought) the first Model III version will find a card inside the package which entitles you to get the enhanced version for the difference in original price of the two versions. *Do not* try to send in the card or order it before we inform you or your store that it's ready. Don't become the next victim of the Radio Shack "Black Hole" where orders for unknown products end up!

Model I

Super news for Model I owners who don't have an Expansion Interface and RS-232. You can get our exclusive new TRS-80 Modem I (26-1172), a special software package (26-1139), and cable (26-3009), and communicate (half-duplex) over the telephone lines through your cassette port!

The Model I is a full originate/answer direct-connect (Bell 103 compatible), FCC-approved telephone modem. It contains a standard DB-25, RS-232C connector as well as a DIN connector for the cassette connection and Color Computer's RS-232 port. It's switchable for 0–300 and 0–600 baud, although baud rate is software selectable. When used with RS-232 connection (rather than Model I cassette port) it's full duplex. Cable is not included in the low price of only $149! Don't wait for this one to show up under another brand name. We designed it, and we build it here in Texas.

We've got a few more tricks up our sleeve for Model I owners in the coming months, so stay tuned!

New Computer Catalog

About the time you read this, we should be getting our new RSC-5 catalog to the stores. The cover looks just like the current RSC-4, but you'll find some new goodies in it. For example, we now have blank labels to fit both 5-1/4 and 8-inch floppy diskettes, an anti-glare screen to go over your video screen, a 15-inch wide stackable data printout tray and a univeral TRS-80 binder you have asked for with pages that will hold cassettes and disks. There's even a logic template to help draw flowcharts.

RSC-5 lists our new Line Printer V, which quietly replaced the III in December. Advancing our technology allowed us not only to speed it up to 160 CPS (from LP III's 120), but to add underscoring, descenders, and a new 7.5 CPI condensed-bold character mode. The character set includes 94 alphanumeric characters, 26 European symbols and 30 graphics patterns. Best news of all, the price is $100 lower than the LP III—only $1,860 (plus cable)!

There simply wasn't room in RSC-5 (or time) to try to "shoehorn" in our new software packages, so don't look for a big change there. We did get Visicalc in for Models I, II, and III. We also included a new Statistical Analysis program for Model II, and Mailing List II (also Model II), which interfaces with Scripsit. There's another new peripheral of special interest to Color Computer owners, but I'll tell you about it next month.

Questions Again

Customers often ask why we don't keep our stores better informed about delivery times and new products. It's true. Often they can't tell you when a new product will be available, or how long the wait will be for a back-ordered item. They receive an update list of temporarily out-of-stock computer items every few weeks. We can't guarantee the dates, but they represent our best (and latest) educated estimate.

Sometimes the problem is non-delivery from our vendor, a transporation problem, a last minute bug, or a "hold" by our incoming Quality Assurance staff. Software shipments virtually stopped for six weeks once, because our binder supplier went on strike.

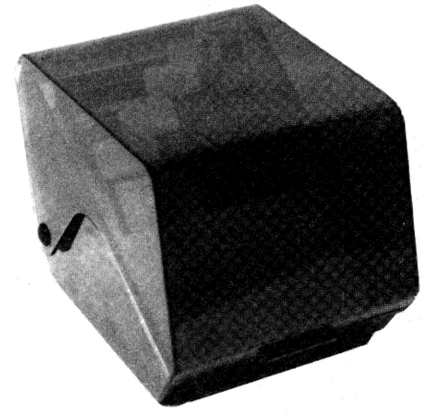
In extremely dry cool weather, static electricity can cause a 20 percent reject rate in duplicated cassettes, and anything over three percent triggers a QA delay on hardware or software. Our belief is that the only thing you appreciate less than not getting your item is getting one that doesn't work!

Continued to p. 37

META TECHNOLOGIES
FOR YOUR DISK SYSTEM

FILE BOX
DISKETTE STORAGE SYSTEM

$24.95 for 5¼" disks
for 8" disks . . . $29.95



MTC brings you the ULTIMATE diskette storage system, at an affordable price. Storing 50 to 60 diskettes, this durable, smoke-colored acrylic unit provides easy access through the use of index dividers and adjustable tabs. Unique lid design provides dust-free protection and doubles as a carrying handle.

PLASTIC LIBRARY CASES
(not shown)

An economical form of storage for 10 to 15 diskettes, and is suitable for your bookshelf! Case opens into a vertical holder for easy access.

5¼-inch diskette case $3.25
8-inch diskette case $3.50

Single Sided, Single Density, Soft-Sectored 5¼-inch, (for TRS-80™) Mini-floppy

DISKETTES
$21.95 box of 10

These are factory fresh, absolutely first quality (no seconds!) mini-floppies. They are complete with envelopes, labels and write-protect tabs in a shrink-wrapped box.

PLAIN JANE™
DISKETTES
The Beautiful Floppy with the Magnetic Personality™

Thousands of people have switched to this low-cost alternative. These quality diskettes are packaged in a plain white box . . . no fancy printing, fancy names or fancy labels, not even our own (labels cost money). Trust us.

PLAIN JANE™ Diskettes $21.95
10 boxes of 10 . . . . (each box)$21.50

VERBATIM'S PREMIUM DISKETTES AT AFFORDABLE PRICES

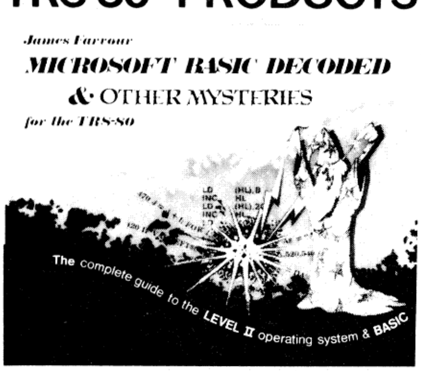
DATALIFE™

Seven data-shielding improvements mean greater durability and longer data life. These individually certified diskettes feature thicker oxide coating, longer-lasting lubricant, improved liner, superior polishing and more! Meets or exceeds IBM, Shugart, ANSI, ECMA and ISO standards. Buy the best . . . buy DATALIFE™.

VERBATIM DATALIFE™ DISKETTES
5¼-inch (box of 10)
MD525-01 . . . . $26.95
10 boxes of 10 . . . . (each box)$25.95
8-inch FLOPPIES
Single-Density, FD34-1000 . . . $29.95
Double-Density, FD34-8000 . . $39.95

MORE PRODUCTS →

TRS-80™ PRODUCTS



NEWDOS/80 by Apparat $149.95
NEWDOS+ with ALL UTILITIES
35-track $69.95
40-track $79.95
TRS-80™ DISK AND OTHER MYSTERIES $19.95
MICROSOFT ™ BASIC DECODED & OTHER MYSTERIES for the TRS-80™ $29.95

'RINGS' & THINGS

Help prevent data loss and media damage due to improper diskette centering and rotation with the FLOPPY SAVER™ reinforcing hub ring kit. 7-mil mylar rings install in seconds. Kit is complete with centering tool, pressure ring, 25 adhesive backed hub rings and instructions.

HUB RING KIT for 5¼" diskettes . . $9.95
REFILLS (50 Hub Rings) $4.95

Protect your expensive disk drives and your valuable diskettes with our diskette drive head cleaning kit. The kit, consisting of a pair of special "diskettes", cleaning solution and instructions, can be used for 52 cleanings. Removes contamination from recording surfaces in seconds without harming drives.

CLEANING KIT for 5¼" drives . . . $24.95

Products damaged in transit will be exchanged. Prices, Specifications, and Offerings subject to change without notice.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

PRICES IN EFFECT February 1, 1980 THRU February 28, 1980

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $2.50 for standard UPS shipping & handling
- $2.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

FOR PRODUCT INFO 1-800-321-3640 — **CALL TOLL FREE** — **TO PLACE ORDER 1-800-321-3552**

IN OHIO call (216)289-7500 (COLLECT)

master charge — VISA

META TECHNOLOGIES CORPORATION
26111 Brush Avenue, Euclid, Ohio 44132

✓20
801219

TRS-80 is a TM of Tandy Corp.
PLAIN JANE is a TM of MTC.

80 INPUT

"Being somewhat of sound mind, I fathomed that a parallel output port would have difficulty driving a serial printer. . ."

Thanks to Friesen

I have rarely seen an article as beneficial as Frank Friesen's "H-14, Meet The TRS-80" in the October 1980 issue.

A year ago, I began searching for a printer. The Radio Shack standard models seemed overpriced for what they could do ("Wait 'til the screen printer comes out" I was told). I had experience with the teletype (all bad), had never heard of Baudot as a printer and the Selectric interfaces had not yet hit the market.

Then along came good ol' Heathkit with their $595 build-it-yourself H-14. Being a veteran of a tv, stereo and several pieces of test equipment, I couldn't see how I could go wrong so I sent a check to Benton Harbor.

Being somewhat of sound mind, I fathomed that a parallel output port would have difficulty driving a serial printer but, having read the RS new product releases, I was aware of the RS-232C interface. My local neighborhood store had one with, lo and behold, an application/instruction manual which described and listed a printer driver.

The kit was a snap. It went together just as the book said it should, and the test switch even printed the full character set the first time it was pushed.

After getting past the problem that both the printer and the RS-232C interface had the same gender interconnects, I loaded the Radio Shack driver, set the baud switches to 110 and entered LPRINT "Hello"—it worked—but oh, so slowly. Further reading showed that I could go all the way to 9600 baud if I wanted (although Heath did not recommend it, but I thought that statement was for the H-8). Come to find out the only speed that I could get, reliably, was 110 baud—how boring!

Then I noticed that Heath was offering a software interface to the TRS-80.

Entered the driver and, what do you know, welcome to the world of 4800 baud.

Everything was fine and dandy until:

How do you list out an assembly language program while in TRSDOS?

If you don't yet know how to use the USR function, where do you put all of the data statements for the BASIC POKE routine, in a program with many data reads?

Since I wasn't into assembly language programming, the first consideration didn't bother me too much, but the second turned out to be a bear of a bug. I spent half my keyboard time trying to figure out where to put the 72 data items required by the BASIC driver so they wouldn't interfere with the body of the program.

After locking up the printer and burning up the print head ($203) I resigned myself to an eternity of fighting the data statement battle.

Then comes Mr. Friesen, with a two-component interface for less than $15.

This one article has paid for my subscription. My computer now has no idea that a parallel printer is not connected to it. When I can print while in DOS I know I have something.

How much is a lifetime subscription? And I will consider it only if you give Mr. Friesen the "Article of the Year" award.

Mike Bloom
Houston, TX 77055

Software Copyright Law

Your November 1980 issue of *80 Microcomputing* contains an error on page 48 in your News section. The following information is correct and might be brought to the attention of readers of your "Software Copyright Law" article.

H.R. 6933 was not passed into law on August 20th. What happened is that an amendment was approved to the draft then within the House Government Operations Subcommittee. Yet another amendment changed the bill again on September 23. As amended, the two most controversial parts of the bill (which would have created an independent Patent and Trademark Office apart from the Commerce Department, and which would have merged the Copyright Office with PTO) were both dropped. As revised, the bill was approved by the full House Government Operations Committee and, the same day, accepted by the House Judiciary Committee. Despite a last ditch effort to get it to the floor of the House before the election, it was delayed and scheduled to come up in the lame duck House on November 12—since then I've not looked it up.

By the way, the first *Data-Cash* wasn't really overruled either—but that's another story . . .

Walter Schmidt
Bronx, NY 10463

We were informed by a member of the CONTU commission and the counsel for the Subcommittee on Courts, Civil Liberties and Administration of Justice, that HR 6933 had passed into law *on August 20, 1980. At the time, we also checked with the Bill Status Office which had no record of the bill, but informed us that it often took several weeks for the information to reach their office.*

Upon receiving Mr. Schmidt's letter, we again contacted the counsel for the Subcommittee; we also contacted the Bill Status Office and the Executive Clerk's Office at the White House. We stand corrected. HR 6933 had not passed into law on August 20, but did indeed pass in the House and Senate on November 21. At press time, the bill had not yet been signed into law by President Carter.

By the way, Mr. Schmidt, we did not report that Data Cash vs. J.S.&A. was over ruled; what we said was that the reasoning behind the ruling had been altered.
—Eds.

Letters to the Shack

I've read Wayne Green's comments regarding the improvement in Radio Shack's relationship to the public. Since I'm a bit cynical in general I was doubtful. On November 13, 1980 I wrote a letter that was loaded with questions and criticism to Mr. Juge. I received the following in reply.

"Dear Mr. Dunn;

Ed Judge has asked that I reply to your letter of November 13th.

Thank you for your constructive comments as well as for the many compliments . . ."

At this point the letter has a page and a half of answers to my questions and problems. These answers are direct and to the point with no issue evading.

"... once again, thank you for all your comments. They are all listened to. I hope that I have been able to provide you the information you needed but if I didn't, please don't hesitate to drop me a line."

This letter is signed by Bill Walters, Consumer Information Manager. As I said, I'm a cynic—but results are results. They solved my problems in a businesslike manner, and fast.

I rely on my two TRS-80 Model I's in my accounting business and in the production of software. They are effective. I'm looking forward to delivery of my Model III. It is important to me that good working relationships exist between me, the magazines I rely on, and the computer company I deal with. Therefore I feel it important to express my appreciation to Wayne Green for getting Tandy to realize that customer relations are important; to Ed Juge for initializing and overseeing this very tangible improvement; and to Bill Walters for doing his job in an excellent manner.

My hat is off to all three of you.

Peter G. Dunn, President
Sturdivant and Dunn, Inc.
Conway, NH

Software Material Needed

A Commodore PET-16K and a Radio Shack TRS-80 16K-Level II were recently placed on loan to the men here at the prison. We would like to develop our skills and the full potential of the computers with our studies in electronics and microcomputer programming. However, we are finding it difficult to do without the appropriate instructional material and software models to guide us. I have talked with our prison librarian and tried to get him to order several texts covering microprocessor programming and software development, but he informed me that the library budget for the coming year has already been allocated, so he would be unable to order any for us.

Confinement at Folsom Prison presents a real problem in obtaining study and instructional materials of this type. We can borrow software textbooks from the California State Library when they are available, but they can be kept for only brief periods of time. The time period we are allowed to keep them is just not sufficient for us to properly utilize the information and instructional aids they contain. Would it be possible to impose upon you to send us a complimentary copy of any educational, business application, or systems program listings or tapes you may have available? (Both our computers have cassette tape I/O.) The condition of the material makes no difference as long as it is useable. The programs, in addition to utilizing the subject matter, would be used to study commercially developed and professionally written software with the hope that by studying well-written programs we will be able to apply the principles to our own compositions.

If you can see your way clear to send us any program listings or tapes they would be most welcome and sincerely appreciated.

Thank you very much for your consideration of this request!

Gottfried R. von Kronenberger
PO Box B-49542
c/o Mr. R. E. Miller,
Supervisor of Education
Folsom State Prison
Represa, CA 95671

This letter was passed on to us by Dr. Robert Biggs of The Microgram Company. Readers? A subscription to 80 Microcomputing *will be on its way to you, Mr. von Kronenberger.—Eds.*

Mad Hatter

My story begins eight months ago, when I purchased *Sorcerer's Castle*, *M-Trek* and *Galactic Blockade Runner*, put out by Mad Hatter Software. When I brought the programs home, I was surprised to find they wouldn't load! I have the Radio Shack cassette mod, Alphanetic's Acu-Data tape digitizer, and Larry Rosen's CLFIX, and I also followed all of the "generic" loading instructions. Paging through the documentation I was glad to see that Mad Hatter would replace faulty recordings.

I sent back the three programs with high hopes, and anticipation of prompt return. However, three letters and eight months later, I failed to get any kind of response.

If this is Mad Hatter's policy towards legitimate customers, I hope you will warn other potential victims.

Paul Leong
Chicago, IL 60659

At last report Mad Hatter Software was out of business.—Eds.

Software Report

In the November issue of *80*, Wayne Green asked readers to send in comments on software they have purchased.

I am particularly concerned with the high reputation Galactic Software has. The two programs of theirs I have tried are, in my opinion, disgraceful. I purchased Mail List on the basis of Reese

Input conts.

80 AID

Quick Fix Column

At the suggestion of one of our readers, Richard Burrow of Mountain View, CA, 80 is seeking some short "quick fix" items that can be shared by all of its readers.

These items will be reviewed by the editors and placed prominently in the letters section, along with 80 Aid and the all too ubiquitous 80 Debug.

We're not just looking for problems here, but solutions—whether they are homebrew or Radio Shack's own. —Eds.

Line Problems

I searched all my back issues of your magazine without finding any articles on the two subjects I'm interested in.

Has somebody written anything on making the TRS-80 Level II display lines of 80 characters? This is its only remaining significant shortcoming. I hope some genius has found out how to do it or will solve this problem soon.

When I run the following line, the answer is .0100002. Who can tell me why and how to correct it? 120 A = 20.01:B = 20:PRINTA—B.

I hope to see some information on these subjects in your magazine soon.

Michael A. Binkhurst
14939 Sylvan St.
Van Nuys, CA 91441

Fowler's review in the February 1980 *80* issue. My comments follow.

Mail List: Totally lacking in human engineering. Cannot be modified by non-skilled users. Major deficiencies in printing test labels and storing special displays.

Versafile: Equally lacking in human interfacing. The user is left with a blank screen—apart from a prompt symbol—as his first item. Requires the user to have a knowledge of LISTing and EDITing—which is not mentioned in the advertising.

To balance this, I have nothing but praise for Radio Shack's ICS Inventory Control System, which I have used as the basis for several general purpose filing systems. The documentation of the program (as opposed to the user's manual) is of course non-existent, but it *is* a good program.

I also have high praise for Express Marketing's Project Schedule Analysis Package PSA/1. The 61 page manual is a gem. Any unskilled user should be able to operate the program without any problems, first time. A real bargain, and easy to customize.

M. Barlow
Pierrefonds, Quebec, H8Z 2A8

Good News

Could I pass on a good word? All too often we read of troubles and woe. We worry when we send off our money orders and hope at least for reasonably quick delivery. Well, I just got two pleasant surprises from Fuller Software. Not only did the Supermap I ordered arrive within a couple of weeks, but it was accompanied by a $9.00 check. It would appear that the price has been reduced. How unusual and comforting to find this sort of integrity—it made my day!

Jock F. McTavish
Whitehorse, Yukon, Canada

Pyramid Still Lives

After reading the large number of letters endorsing the Radio Shack program "Pyramid," I was thoroughly infuriated. I have spent at least three days in frustrated adventuring and have, needless to say, given up. This sort of program must be intended for people with a good knowledge of assembly language (to cheat), or for very experienced adventurers.

When I first bought my 80, I bought Pyramid as a starter program. I have not found my way through yet. Many find this sort of thing fun, but I come to the conclusion that it is not a beginners delight, as letters in the last issue imply. I think it is a vague and over-acclaimed mess.

I will probably continue trying, for when (and if) I find my way out alive, I will *really* have something to tell my grand-kids.

True, the idea of calling the writer of a program is very brash, and the review was dastardly. The author made a good point however—just because you are smart enough to get through the damned maze doesn't mean that every CLOADing freak in TRS-80 land should waste $14.95 to challenge your score.

Glenn Mcgee
President GMINC Software
Waco, TX 76706

LNW Expansion Board

I recently purchased a System Expansion PC board from LNW Research. My first application was to interface an H-14 Heathkit Printer to my TRS-80. I ran into a few problems that might interest other users. Since I have not yet expanded my board beyond this application, response from other users would be helpful.

Looking at the schematic, I noticed that the terminating resistors for D4 through D7, CAS, RD and MUX were 1k ohm/220 ohm pairs. I would have expected these resistor pairs to be nearly equal and also noticed that D0–D3 was terminated with 430 ohm/220 ohm pairs. I changed all the terminating resistor pairs to 430 ohm/220 ohm. This resolved the hanging problem.

The H-14 printer connection to the expansion board (if wired by the expansion board schematic) connected input to input and output to output. To resolve this, connections 4 and 5 were swapped and 6 and 20 were swapped in the DB 25 connector.

The H-14 busy signal is RTS pin 4 (pin 5 in the expansion board). This signal must be inverted and fed to the parallel printer interface. (The TRS-80 addresses the parallel printer in this configuration.) A jumper from U 52 pin 3 to U 20 pin 3 and U 20 pin 4 to U 32 pin 11 (the remaining jumpers are as specified by LNW) will accomplish the above.

Harold L. Drurey
Westerly, RI

Pocket Computer

I have been very impressed by Radio Shack's new product line. The Pocket Computer, especially, appears to be a very useful addition to their already outstanding collection of inexpensive computer products.

For the past several months, I have been looking for a device which I could use in a remote data collection application. Indeed, I have gone so far as to have a unit designed and prototyped which would enable me to enter data in the field, and upload that data to a large minicomputer system for subsequent processing. The Pocket Computer, when announced, appeared to be capable of performing all of the requirements of my data collection system at a fraction (1/20) of the cost, and I was overjoyed.

Unfortunately, the Pocket Computer can't talk to anything but itself. The Pocket Computer is capable of writing to cassette tape, but the tape format is unique unto the Pocket Computer! Radio Shack is aware of this minor deficiency, yet no solution has been offered.

Do any of your readers have a program for Model I, II, or III TRS-80 which will read a cassette tape produced by a Pocket Computer? I would prefer not to have to wait for Tandy to write the program, and for Apparat to fix it.

Rick Richmond
621 West Fontanero
Colorado Springs, CO 80907

Variable Change

The "Variable Documentation" article and program by William Noel in your September issue have been very useful. There were several changes that I had to make for it to work with disk BASIC under TRSDOS 2.3 using 48K RAM:

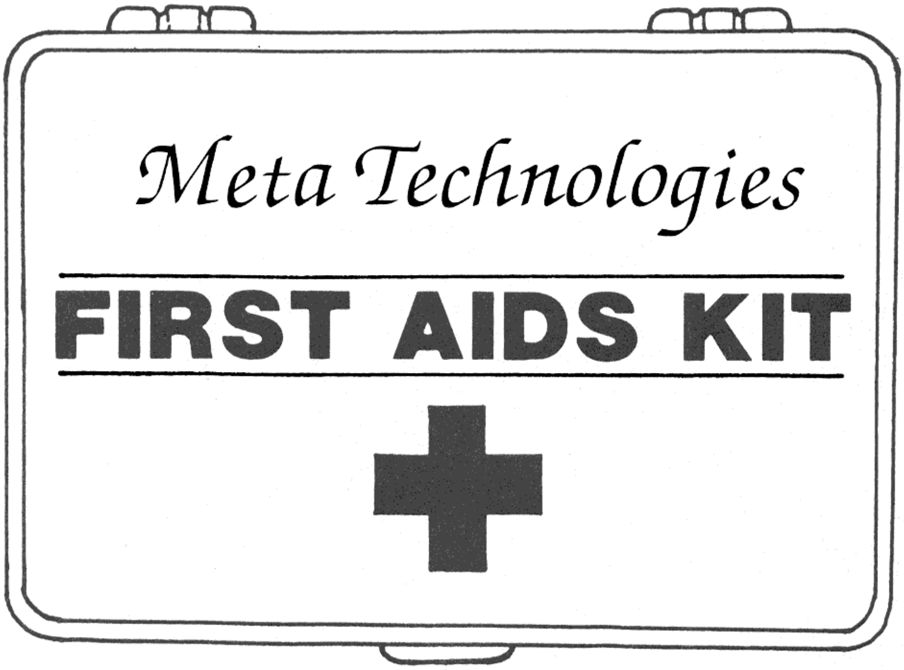
1. In line number 65010, change address 17129 to address 27172 (not 26302).
2. In line number 65012, change address 17128 to address 27171 (not 26301).
3. In line number 65020, change address 17129 to address 27172 (not 26302).
4. Convert line 65280 into the following two lines to express addresses between decimal 32768 and 65535 as decimal −32768 through −1:

```
65280 V = 27172 + U:IF V> = 32768 THEN 65290
ELSE
      T = PEEK(27172 + U):U = U + 1:RETURN
65290T = PEEK(27172 + U − 65536):U = U + 1:-
      RETURN
```

Doug Walker
Salem, OR

Input conts. to p. 31

AILING INFORMATION ?



DOCTOR IT UP WITH MTC AIDS-I™

WORKS WITH CASSETTE AND DISK!

- Up to 10 user-defined fields, records up to 254 characters long.
- Machine-code assisted sorting: 200 records in about 5 seconds, any combination of fields.
- Full editing capabilities:
  Backspace, Right-justify, Delete field, Restore field, Skip field, Enter field, Skip record, Delete record.
- Full selection capabilities:
  Choose records to be worked on using any one of 7 comparisons.
  Examples: NAME greater than L or STATE equal OH or PRICE less than 99.00
- Selections effective for the following main functions:
  - LOAD records from cassette or disk
  - SAVE records to cassette or disk
  - SORT records
  - DELETE records
  - UPDATE/ADD records
  - PRINT/DISPLAY records
- Print/display any combination of fields in any order, in any position on a page-use for mailing labels, lists, etc.

MTC AIDS-I (Model I) $34.95

Write for our complete catalog, or

CALL TOLL FREE

FOR PRODUCT INFO 1-800-321-3640

TO PLACE ORDER 1-800-321-3552

IN OHIO call (216)289-7500 (COLLECT)

master charge

VISA

META TECHNOLOGIES CORPORATION

26111 Brush Avenue, Euclid, Ohio 44132

✓20

801219

TRS-80 is a TM of Tandy Corp.

✓ Reader Service—see page 242

EDUCATION REVIEWS

Edited by Pamela Petrakos

"This close involvement of man with machine will, in Papert's view, yield some startling results in the areas of learning, education and socialization."

Mindstorms: Children, Computers and Powerful Ideas

Seymour Papert
Basic Books, Inc.
New York, N.Y.
Hardcover, 230 pp.
$12.95

by Chris Brown
***80* Staff**

Mindstorms, by Seymour Papert, is a revolutionary book about what can and might be done with computers in the field of education. Its content is drawn from the author's years of research and bears little relation to the drill applications that proliferate in computer aided instruction (CAI).

Two central themes occur in *Mindstorms.* The first is that children can learn to use computers in a most powerful way. The second, is that learning to use computers can change the way children learn everything else in our society.

Early in his introduction, Papert candidly states, "It is not true to say that the image of a child's relationship with a computer I shall describe here goes far beyond what is common in today's schools. My image does not go beyond: It goes in the opposite direction." And so it does.

Microworlds

In *Mindstorms,* Seymour Papert gives us a view of educational computing not available anywhere else. The vision he conjures of the computer's role in education in the next decade finds the machine and the student in symbiosis: The student controlling the machine (not vice versa), and in doing so exercising control over an exceptionally rich and sophisticated microworld. This close involvement of man with machine will, in Papert's view, yield some startling results in the areas of learning, education and socialization. One of the most tangible will be a greater understanding of the abstract in the very young. The author even feels that we can overcome our inherent mathematical phobia with the conceptual help of computers.

Papert's theoretical roots are found in the teachings of Jean Piaget. In Papert's words, "I take from Piaget a model of children as builders of their own intellectual structures. I call Piagetian learning, learning without being taught." Papert expands this model, however, by saying, "But in many cases where Piaget would explain the slower development of a concept by its greater complexity or formality, I see the critical factor as the relative poverty of the culture in those materials that would make the concept simple and concrete."

This is where the computer comes into play for Seymour Papert. Computers can supply children with examples and models which simplify those complex concepts. Papert's LOGO language, the result of ten years research at MIT, is the key that will unlock the door to these rich conceptual microworlds within the computer.

LOGO is actually a group of symbol-oriented, interpretive languages that include procedural definitions as well as the possibility of recursion. Using LOGO, young children can program cybernetic robots, called Turtles, to draw patterns or follow instructions. By programming in symbol-oriented "Turtletalk," students are able to participate in the creation of Turtle geometry. This intimate level of participation, in Papert's view, provides an understanding of the abstract concepts underlying most advanced learning.

Theories at Work

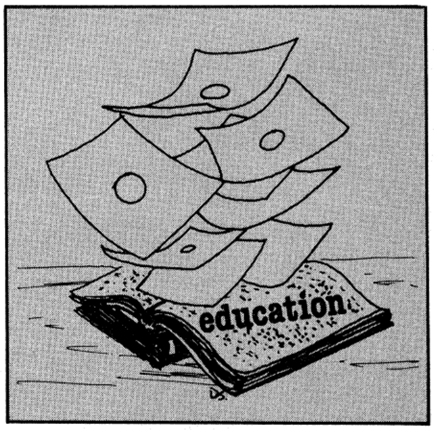
Mindstorms provides many examples of Papert's theories at work. To understand the nature of a differential, students are asked to program their Turtles to draw a circle. To do so, the student must instruct the Turtle to move forward, then turn a bit and move forward once again, repeating the instruction until a circle is drawn. The lack of continuity in the Turtle's action that eventually results in a circular figure illustrates the nature of the differential. The student can grasp the concept of a differential without being formally taught the intricacies of calculus by simply participating in its creation. More importantly, a student can grasp this concept at a much younger age. The chapters of *Mindstorms* are rich in such examples of the techniques that Papert has developed during the course of his LOGO experiments.

New Horizons

The author maintains a balance between the purely theoretical aspects of his work and the practical results he sees in his lab. In addition to being a lucid account of innovative research, *Mindstorms* is also a well written book (something rare in research accounts). Papert's fluid style draws the reader on to new horizons in educational computing and each chapter telegraphs to the reader the author's own excitement about what he has accomplished to date and what he hopes will be accomplished in the future.

Refreshing and Innovative

In *Mindstorms* Seymour Papert gives us a view of computers in education that is refreshing as well as innovative. His ability to look at the machine in a new way, to visualize applications that have never been thought of, to look beyond what is being done now, to what might be done tomorrow, is truly exciting. In *Mindstorms* he shares his vision with us. It is required reading for anyone with an interest in computers, children, education or the future. I recommend it highly! ■



Curvplot $16.95
Curvfit $16.95
MTS Enterprises
Niceville, FL

by Scott Spangenberg

Worthwhile books are an enjoyable experience. They provide valuable information and insights, making the reading experience entertaining. Others yield their treasures in a more sullen fashion. The same is also true of programs. Of these two pieces of software, I found Curvplot entertaining and informative and Curvfit to be of value as a teaching tool, but not as exciting to use.

Both tapes loaded easily, and the signals were clear and strong. Likewise, the quality of programming was excellent in both cases.

Curvfit

Curvit is a polynomial curve fitting program which will handle up to 14th degree polynomials and 40 data points. A unique feature of the program is that it will allow you to input up to four y values for each of the x values. The program will then generate a separate curve fit for each set of data. The program takes between 20 seconds and a little over half a minute for a third or fourth degree fit.

A 14th degree fit with 40 data points will take better than 10 minutes. The program will calculate the coefficients for each power of x in the selected model, predicted values of y over an interval, and of course the correlation coefficient. You are also presented with the option of analyzing the same data set for a different degree of fit (a nice touch), and the program will display the correlation coefficients for all models chosen.

My major gripes are: The instructions within the program are a little dense and awkward, although you can figure out what you are being asked to do without wracking your brains. (The documentation is a great deal more lucid and would be particularly helpful to those not familiar with the pitfalls of curve fitting.) Second, I would have preferred a whole family of curve fitting programs, not just a polynomial curve fitting. For example, exponential and power functions occur quite frequently in business data.

My only other gripe is a minor one. Commands phrased as questions (such as "HIT ENTER TO CONTINUE?") are confusing to the new computer user. This problem could have been eliminated by using PRINT to display the prompt and INKEY$ to read the response.

Curvplot

Curvplot also has the same bug, for example, "HIT ENTER TO SEE MENU OF COMMANDS?" However, the documentation for this program was also very well done, although I did find some minor mistakes, such as misspellings.

This program is a lot of fun to use and would be excellent for teaching, because it would allow students to discover the shapes and fundamental natures of functions. Any function which may be estimated by either the explicit functions provided in TRS-80 BASIC or any approximation algorithm that BASIC can handle can be plotted using Curvplot.

The program will automatically send you into TRS-80 edit mode to place your function into the body of the program; an original feature. You may change the coarseness of the plot, draw only the axes, relabel the axes if your plot overwrites any of the labels, and refer back to the menu of commands at any point in the program. The user must define the range of values that are to be plotted.

One of the most useful features of the program is that you may also change the range of x or y values that are to be plotted. This feature also allows you to zero in on any areas of special interest.

It's possible to make the program bomb if you ingnore the instructions, but basically, both programs are essentially user-proof.

I consider both of these programs to be very useful. However, Curvplot is actually fun to use whether you are a student, a lover of math or just plain curious. ■

Scott Spangenberg received a B.S. in math, with a minor in physics, from the University of New Hampshire. He has been tutoring math for ten years and also teaches programming.

Part I: Introduction to BASIC
Tandy/Radio Shack
Ft. Worth TX
$159

Package Includes:
Teacher's Manual
25 Student Workbooks
143 Transparencies

by Joan Poltack

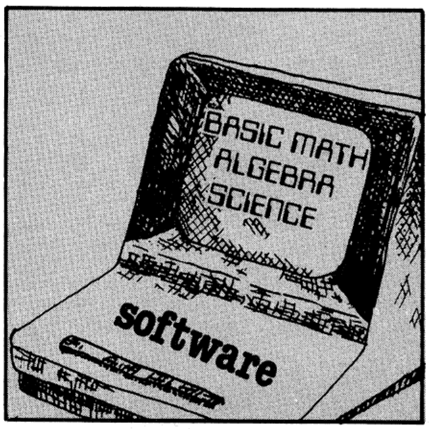
Part I: Introduction to BASIC is a complete package of teaching materials consisting of overhead transparencies (143 in all), a teacher's guide and student manuals. All of this is conveniently contained in a three-ring binder. The guide has been designed for use by any teacher, regardless of his/her experience with microcomputers.

The binder is designed with pages one above the other. The upper page directs the teacher so explicitly that, indeed, nearly anyone could teach the lessons with little or no advance preparation. On the lower page the teacher finds an exact reproduction of the overhead transparency he is to show.

Pages in the student manual are closely related to the transparencies. The student is directed to turn to specific pages and is told when to complete blanks left in his manual. The answers are found in the back of the student manual. Each lesson culminates in a hands-on activity with the computer and a chance to test a student's new knowledge. If a student has mastered his lessons, then the activities are within his grasp. The student manual and teacher's guide each end with a section called "Feedback for Activities" which shows the complete, correct programs.

Part I: Introduction to BASIC has been extensively field tested on students ranging from elementary school age to college professors. Taken as a whole, it's a most successful package that can give a group of computer novices the confidence and knowledge to delve further into programming on their own. ■

Joan Polteck received a M. Ed. from Boston College. Currently, she is the Media Specialist at Martha Jones School in Westwood, Massachusetts.



K-8 Math Program
Tandy/Radio Shack
Ft. Worth, TX
$199

Package Includes:
Teacher's Manual
10 Different Programs
5 Cassettes
3 Diskettes

by Frances H. Petrakos

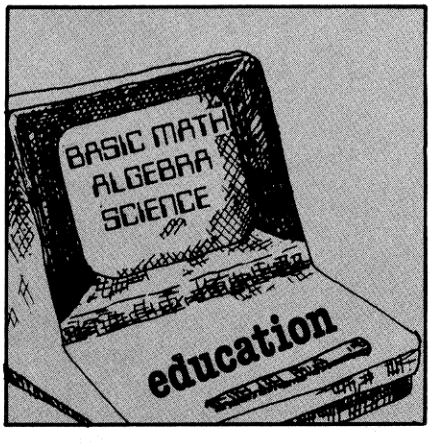
Research has identified poor recall of math facts as a major source of students' skill deficiencies. It's a sad fact that once students fall behind in arithmetic, they tend to stay there.

Radio Shack's K-8 Math Program is an excellent computer supplement to regular instruction that can be used successfully, with either the above-average student who wishes to sharpen his skills or the average and below-average student, who requires a remedial drill to increase his skill.

Excites all Learners

Observing students work with this program, one thing is clear—K-8 Math Program excites all types of learners. It packs the kind of interest and variety that children love—along with the kind of solid instruction that effectively teaches computation skills. Drill work is of primary importance in arithmetic instruction and students get plenty of it here. But these are drills students want to do. It's one thing for a student to be faced with a page of addition problems, but it's something else when the computer tells the students which column to add first and whether there is regrouping or borrowing and where to put that sometimes illusive number.

Each feature of the program is success-oriented. They are as follows:



- Automatic promotion of the student, based on a minimum number of problems and a minimum or maximum score.
- Placement mode to move the student ahead or back very rapidly to an appropriate level, where regular drills begin.
- Reinforcement message keyed to correct and incorrect answers.
- For grades K-3, reinforcement messages use graphics. Smiley, the happy face, is the student's cue that he has answered the question correctly. Later grades were reinforced with statements like "Great", or "That's correct!"
- A unique student time-out function prompts him. If a student doesn't respond after 15 seconds, one of a series of prompting messages tells him to "Come on, try one," or "Don't go to sleep."
- Session that can be set to run a specific number of problems and terminate automatically, or can be ended after a time limit by using a keyword known only to the instructor.
- Individualized instruction for each student. Each student moves at his own pace, constantly challenged, but never threatened. This is certainly important, if the program were being used by a remedial student.
- A flashing cursor denotes borrowing in subtraction or carrying a number in addition. This same procedure is used for multiplication and division problems. This is a very good technique for the slower than average student, who has difficulty remembering the steps in the correct sequence.
- A comprehensive reporting function is provided at the end of a student session. The screen displays the total number of problems attempted, number correct, percent correct, any promotions and average response time. All these features offer learning incentives to the student.

The package also includes a Student Record Summary Sheet, which is filled in at the end of each session. The teacher, with a quick glance can see exactly where a student's strengths and weaknesses lie. This automatically shows a teacher or aide what lesson to continue at the next session. Thus the automatic promotion or demotion feature allows the students to progress at their own individual rate.

Skill Building Lessons

The skill building lessons in addition, subtraction, multiplication and division present more difficult problems as the lesson numbers increase. An appendix outlines the problems covered within each lesson, thus making it easier for a teacher to place his/her student. A teacher can select the number of problems he/she wants each student to work, and can also time the sessions.

"Encounter after encounter with success motivates a student, and when a program doesn't 'feel like math'. . . then indeed it is a positive learning experience."

The program's built-in features of immediate feedback of correct or incorrect problem solving make it an invaluable learning tool. The criteria for evaluating a student's progress are also built into each program. For example, if a student is having difficulty, he is required to work additional problems and must solve eighty percent of the problems before he is promoted to the next level. If the student is unable to work at least forty percent of the problems correctly, the previous lesson is repeated.

K-8 Math Program is an excellent teaching tool. The tapes for each grade level begin with the most basic arithmetic facts and computational skills, so that no matter how far behind a student may be, he or she won't be left out. And you can appropriately place the better student, offering him a challenging, stimulating experience.

The math level in the tapes is carefully controlled to make the program accessible to students of mixed abilities. And because only one math skill is covered in each lesson, students practice intensely and acquire the reinforcement they need to master that skill before they move on to the next.

Radio Shack's program is designed for simplicity and ease of student use. Encounter after encounter with success motivates a student, and when a program doesn't "feel like math" (as expressed by one student), then indeed it is a positive learning experience. ■

Frances Petrakos received an M. Ed. from Boston State College. She has been teaching for the past ten years on various levels—from grammar school to high school. Currently, she is a teacher of moderate special needs in Westwood, Massachusetts.

META TECHNOLOGIES

CALL TOLL FREE 1-800-321-3552 TO ORDER

MTC AIDS-III™

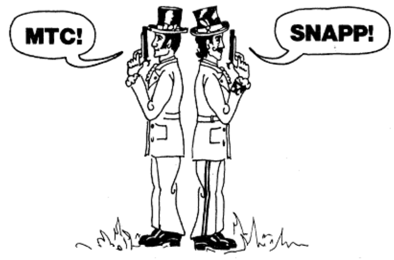
MODEL I . . .$69.95 MODEL II . . .$99.95

Introducing the latest addition to MTC's family of data management systems, AIDS-III. NO PROGRAMMING, easy to use. COMPLETE PACKAGE including demonstration application, documentation and MAPS-III (see below).

- Up to 20 USER-DEFINED FIELDS of either numeric- or character-type.
- CHARACTER-type fields may be any length (total: up to 254 characters).
- NUMERIC-type fields feature automatic formatting, rounding, decimal alignment and validation.
- Full feature EDITING when adding or changing records:
  - ENTER FIELD (can't type-in more characters than specified).
  - BACKSPACE (delete last character typed).
  - DELETE FIELD contents.
  - RESTORE FIELD contents.
  - RIGHT-JUSTIFY FIELD contents.
  - SKIP FIELD (to next or previous field).
  - SKIP RECORD (to next or previous record).
- SORTING of records is MACHINE CODE assisted.
  - 200 RECORDS (40 characters) in about 5 SECONDS.
  - ANY COMBINATION of fields (including numerics) with each field in ascending or descending order.
- SELECTION of records for Loading, Updating, Deleting, Printing and Saving is MACHINE CODE assisted.
  - Specify up to 4 CRITERIA, each using one of 6 RELATIONAL COMPARISONS.
  - LOAD or SAVE selected records using MULTIPLE FILES.
  - Example: Select records representing those people who live in the state of Colorado, but not in the city of Denver, whose last names begin with "F" and whose incomes exceed $9000.00.
  - Example: Select records representing those sales made to XYZ COMPANY that exceed $25.00, between the dates 03/15 and 04/10.

MAPS-III (MTC AIDS PRINT SUBSYSTEM), included at no charge, has the following features:

- Full AIDS-III SELECTION capabilities.
- Prints user-specified fields DOWN THE PAGE.
- Prints user-specified fields in titled, columnar REPORT FORMAT, automatically generating column headings, paging and (optionally) indentation.
- Can create a single report from MULTIPLE FILES.
- Prints user-defined formats for CUSTOM LABELS, custom forms, etc.



MTC is proud to announce MTC EXTENDED BASIC for the Model II, by R. Ryen. Features include "fixes" to existing BASIC, multi-line functions, extending an existing sequential file, PEEK, POKE, greatly enhanced screen control and expanded editing capabilities. The contents of variables are NOT CHANGED when editing, deleting, inserting or merging lines, allowing continued program execution! All this and much more. Compatible with SNAPP BASIC, below.

MTC EXTENDED BASIC . . . . $ 99.95

MTC brings you the best of SNAPP, Inc.'s Model II BASIC interpreter at a very special introductory price. Written entirely in machine language, the enhancements are fully integrated into BASIC and require no user memory or disk space. Utilizes APPARAT's NEWDOS modifications to BASIC on the Model II. Features include 16 single keystroke commands for editing, listing, and other operations. An enhanced program line renumbering facility supports relocation and duplication of blocks of code. Includes a powerful cross-reference capability for producing a list identifying program line locations of user-specified variables and line numbers. Output may be displayed or printed. Compatible with MTC EXTENDED BASIC, above.

SNAPP BASIC for Model II . . . . $ 99.95

MTC AIDS CALCULATION SUBSYSTEM-III™

MODEL I . . .$24.95 MODEL II . . .$39.95

MTC's most popular AIDS subsystem. Use for report generation involving basic manipulation of numeric data. Features are:

- User-specified page title
- Columnar Headings
- Optional Indentation
- Use for accounting, inventory, financial and other numeric-based information systems.
- Columnar subtotals generated when there is a change in a user-specified column.
- User-specified Columnar Totals
- Columnar values computed using constants and/or column values
- Balance forward calculations (Ex: Gross sales equals previous gross sales + sale amount + sales tax).

Compare AIDS-III™/CALCS-III™ with any other data management package under $100! Others make claims, CALCS-III™ delivers!

CALCS-III™ REQUIRES THE PURCHASE OF AIDS-III™ OR AIDS-I™

MTC AIDS MERGE-III™

This subsystem will combine up to 14 AIDS-created data files into a single, large file. An optional purge capability removes duplicate entries while performing the merge operation (can even be used to eliminate duplicates in a single file). Machine-code assisted for high-speed performance, MERGE-III™ properly handles files sorted by any combination of fields, including numerics, with each field in ascending or descending order.

MTC AIDS MERGE-III™ $19.95
For Model II $29.95

MORE PRODUCTS

Let your TRS-80™ Teach You

ASSEMBLY LANGUAGE

REMSOFT's unique package, "INTRODUCTION TO TRS-80® ASSEMBLY PROGRAMMING" includes ten 45-minute lessons on audio cassettes, a display program for each lesson providing illustration & reinforcement, and a text book on TRS-80® Assembly Language Programming. Includes useful routines to access keyboard, video, printer and ROM. Requires 16K - Level II, Model I.

REMASSEM-1 . . . . $69.95
FOR DISK SYSTEMS $74.95

Let Your TRS-80™ Teach You

ASSEMBLY LANGUAGE DISK I/O TECHNIQUES

REMSOFT does it again! REMDISK-1 is a concise, capsulated supplement to REMASSEM-1. Package consists of two 45-minute lessons on audio cassettes, and display programs providing illustration and reinforcement. Provides specific track and sector I/O techniques, and sequential and random file access methods and routines.

REMDISK-1 . . . . $29.95

Let Your TRS-80™ Test Itself With

THE FLOPPY DOCTOR & MEMORY DIAGNOSTIC

by THE MICRO CLINIC

A complete checkup for your Model I. THE FLOPPY DOCTOR completely checks every sector of 35- or 40-track disk drives. Tests motor speed, head positioning, controller functions, status bits and provides complete error logging. THE MEMORY DIAGNOSTIC checks for proper write/read, refresh, executability and exclusivity of all address locations. Includes both diagnostics and complete instruction manual.

SYSTEM DIAGNOSTICS $19.95

EDUCATION REVIEWS

Money Master
Med Systems Software
Chapel Hill, NC
$9.95

by Sherry M. Taylor

There's a new education program from Med Systems Software for the TRS-80. You remember your TRS-80—the one you bought for yourself, but the kids have taken over? If they're going to dominate it, they might as well learn something.

The program is called Money Master and is designed to give children practice counting money. There are drills in adding coins and bills and drills in making change.

As in any good educational program, the student thinks he/she is playing a game. The student is confronted with a randomly generated maze stocked with creatures and objects. To move through the maze, the student must buy the objects and pay a toll to the creatures.

When the student enters a room in which there is an object, the screen clears and shows drawings representing the coins, penny through half-dollar, in a column with a graphics $1 and $5 bill. The student pays for the object by indicating how many of each coin or bill are needed to make up its exact price.

When a creature is encountered, the student is told how much the toll is and what the creature took. Alongside the coins, the student is then shown his change. The student must determine whether the creature gave back the correct change, and indicate yes or no.

If the student's answer is wrong, the computer displays the correct answer. The creature or object is moved to another room to be faced again. If the student gets the amount correct, but another combination of coins is better, that answer is also shown. He is given credit for a correct answer, but cannot use 68 pennies to pay for something that costs 68¢. In this case, the computer informs him that the amount is correct, but he has used too many coins.

Each game is different. There are two dozen creatures and objects stored in memory, but only six are used for any game. The arrangement of the rooms and hallways is random.

A graphic representation of the player shows him walking along the hallways, commanded by the arrow keys on the keyboard. The animation routine is simple, but adds a nice touch.

Although this program is designed for children with some knowledge of money and addition, with the help of a patient tutor even younger children enjoy it. My five-year-old moves through the maze while his older friend works the problems.

When all the objects have been bought and the creatures dispelled, the game ends. Everything the child bought is shown on the screen, along with the percentage of correct answers.

The program has three levels of difficulty. Easy deals with amounts under $1; moderate, up to $5; hard, up to $10.

I wish the program had a preschool level. This option could allow 79 pennies to be accepted for payment, or keep the prices to a total of one coin. This way the preschooler could match the coin to the price, penny by penny.

The program is written in BASIC and uses 16K memory. ■

Vector Addition Program
Tandy/Radio Shack
Ft. Worth, TX
$29.95

Package Includes:
Programmer's Guide
User's Guide

by Mary Shooshan

A good teacher is always on the lookout for good—I mean *really good*—teaching programs. A new addition to Radio Shack's eductional software is Vector Addition, available on cassette or disk. Useful in geometric and physical applications, Vector Addition will find the resultant of up to twelve vectors and display them tip-to-tail or from a common origin.

The program assumes you know what a vector is and how to add vectors by tip-to-tail and parallelogram methods. The manual does go through some examples, but you are responsible for knowing any necessary formulas, e.g. for velocity, force and momentum.

General Instructions

The 47-page manual begins with general instructions on using the TRS-80. This introduction will enable people unfamiliar with the computer (such as students and teachers in a classroom) to get started with a minimum of difficulty.

A very worthwhile section suggests several ways to use computers in school: in the classroom, in a lab, in the library. It compares cassettes to disks as a medium for storing programs. Finally, it notes several things to keep in mind when choosing a location for a computer system.

While the manual does a fairly good job explaining most of the program's special commands, the very useful edit mode, E, is barely mentioned. Don't let this stop you, as editing or altering any of the vectors is easy—just indicate which vector you want to change along with the new magnitude and direction.

The manual goes through one simple example step by step. If you are ever wandering around city blocks that are in perfect alignment with the compass and are curious about possible flight paths for crows, this example is for you. Two more difficult problems with their solutions are also presented.

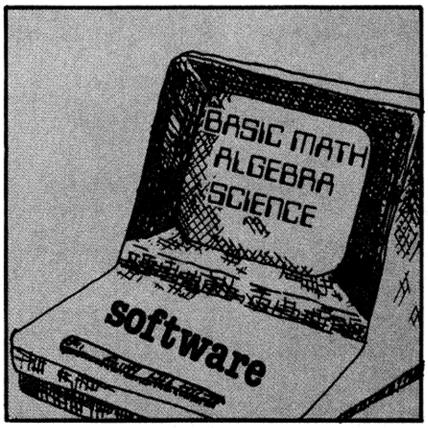
A good listing of problems with solutions is provided—problems covering fundamentals of vectors, displacement vectors, velocity and acceleration, force, gravitation, conservation of momentum, and electric forces and fields—a broad range applicable to both math and science courses.

A Variable Listing

For those who are interested in programming, a guide lists the variables and subdivides the program into sections. Tandy suggests no modifications, however.

How does this program add up? It's well written and uses a good (for TRS-80) graphic display. The manual is well organized for classroom use. This is a worthwhile package for demonstrating vector addition as well as providing student practice. ■

Mary Shooshan received a B.A. from Colby College in math and philosophy. After two years of teaching high school math and science, she retired to pursue educational, as well as other, computer applications. Currently, she is an editor with Instant Software, Inc.



80 REVIEWS

Edited by Pamela Petrakos

"So that you don't get lost in a mind-boggling sequence of computer programming events, here is a beginner's guide. . . ."

Your First Computer
Rodnay Zaks
Sybex, Inc.
Berkeley, CA
Softcover, 258 pp.
$7.50

by Carolyn A. Straub

So that you don't get lost in a mind-boggling sequence of computer programming events, here is a beginner's guide to help you take stock of your business or personal computing needs and plan for your first micro.

Your First Computer, by Rodney Zaks, comes complete with summaries and tests at the ends of chapters along with six appendices of additional, definitive information at the book's end. As a novice, I found it a good foundation.

The figures and photos are clear. In Chapter Seven, on "Business Computing," there is a page by page illustration of a run through a mailing list program. Called NAD (developed by Structured Systems, Oakland, CA), the program has six modules—four are demonstrated in print. The program is written in CBASIC, a compiled version of BASIC. It includes file creation and sort routines in the selection of entries for a typical mailing list. The beginner sees the display step by step on the video screen and is aided by the book.

The author is a bit brief in describing business applications. More detailed examples of business programs which your computer could run might be shown.

Your First Computer takes you in logical progression from the basic configuration of a micro layout to programming, the various computer languages, peripherals, and system applications. Chapter 13, entitled "Help," is a listing of clubs, magazines and other educational outlets where you can go to find out more about your particular area of interest.

The reader is lured into the book by a humorous opening chapter where a family is living within a completely computerized home. By setting these scenes, the author teaches us definitions. The ensuing chapters take us through the micro and its basic components, and there are self-testing exercises (with answers on the next page). From there, we learn how a system works, about programming, flow charting, information representation and applications. Several trademark computers are described briefly, in summaries.

One thing I felt lacking was a thorough discussion of software. Also the final chapter, "Tomorrow," too succinctly and, without much analysis, describes the future of computers.

There is still much to learn about the science behind the operation of the microcomputer. This book is, at least, an incentive to learn more.■

A SAFE House Electronically
Tandy/Radio Shack
Ft. Worth, TX
Softcover, 125 pp.
$1.95

by Bruce R. Evans

Many of you may be wondering how to develop home security systems using your microcomputer. And although you may be able to write the software and build the interface; do you know what hardware is on the shelf or how to assess it?

Well, Radio Shack has come to the rescue with a small book written in distinctive style by Texas Instruments.

Chapter One is an anecdote of the need for protection, which progresses to a narrative on how security has been achieved. Don't let the childish style of this chapter put you off. Just skip it!

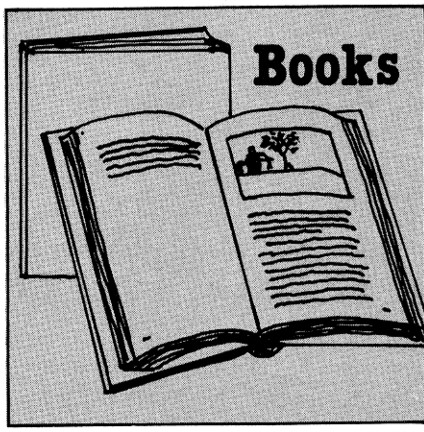
Chapter Two is an overview of the current systems. Although there are photos of some of the equipment—available at Radio Shack—the central thrust gives you the general details of what to buy off the shelves including the limitations. This is a good start for your own system.

Chapter Three gets down to the resistors and capacitors of a system. With three solid examples, the book describes the best way to monitor and protect specific layouts. The pros and cons of each method are spelled out along with the mechanics of the hardware. The simplicity is amazing. Perimeter as well as space systems including infra-red, sonar and microwaves are discussed. The emphasis is on home security to safeguard the house and contents in the owner's absence and, more importantly, to protect the occupants when they are home.

Chapter Four does the same for a business. Here, the emphasis is on detecting and catching an intruder as well as protecting your business from sudden flooding or fire. Most of the ideas presented within this chapter are also applicable to your home system.

Chapter Five deals more explicitly with the hardware, particularly the interconnections that enable response. A few hints are given on using a computer but these are suggestions rather than specifics. I don't feel that this is a failure of the book—it is beyond its scope and purpose.

The book is well-written and factually accurate; I recommend it to anyone who is contemplating building a security system.■



Typing Tutor
Microsoft
Bellevue, WA
$14.95

by Richard S. Adcock

So, you're great with the computer: You can compose a BASIC program in your head without giving it a second thought. But when you get ready to enter the program into the computer, you sit there for an eternity hunting and pecking your way around the keyboard, cursing yourself for having taken basketweaving rather than that typing course.

You could call a local college and see what they offer in the way of typing courses, or you could spend a mere $14.95 for a program recently introduced by Microsoft and, as their brochure suggests, "turn your computer into the Aristotle of Typing Tutors."

Typing Tutor comes on cassette and makes your TRS-80 a top notch typing teacher, capable of improving the typing skills even of a fairly adept typist.

Typing Tutor is designed with the programmer in mind. During the typing exercises, it is not unusual to come across words like DEFINT or DEFDBL which may seem like gibberish to many people, but are of interest to programmers.

Peeking While Pecking

According to Microsoft, the unique thing about this program is a feature they call Time Response Monitoring (TRM), which allows the computer to scan the keyboard 20 times a second, keeping up with even an extremely fast typist. This is also the feature that presumably separates the teachers from the Aristotles. TRM is fast enough to catch even the slight pause between keystrokes which normally indicates that you have peeked at the keyboard.

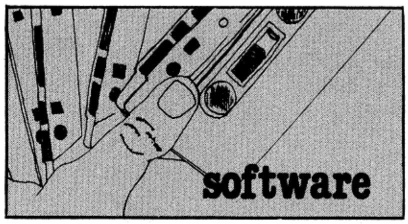
You may choose to practice letters, numbers or symbols, then choose either a practice paragraph or typing lesson. The practice paragraph is used primarily as an indication of current skills. When finished, the computer analyzes the results, indicating speed, accuracy, the characters you were slow on and the characters you missed.

The computer will also instruct you, by giving you different combinations of eight characters to practice. You start with those found on home row, but as you achieve a certain amount of speed (20 wpm starting out) with any one character, it is moved off the practice list and another character takes its place.

The beauty of this system is that you work only with those characters that you need to practice. Old Mrs. Markel, the high school typing teacher, could never have done this.

After you have run through a series of ten lessons, the program allows you to choose a faster, slower or unchanged minimum speed for the next lessons, or will let you try another practice paragraph. The practice paragraph is now composed of those letters, numbers and symbols you practiced until they were removed from your list. You get immediate feedback on how well you're doing in terms of speed and accuracy.

Look at it this way. If you are trying to speed up your computer, a cassette to disk upgrade will improve speed by a fac-



tor of maybe 50 percent. Yet, a slow cassette isn't all that painful, is it? You can always drink a cup of coffee (or sometimes two) while waiting. If you can upgrade your typing speed from a hunt and peck 10 words per minute to 98 percent accuracy at 52 words per minute, a program that you would normally load in five painful hours could be loaded in an hour. Typing Tutor may well be the best deal in a speed-up kit you'll ever buy. ■

Deathmaze 5000
Med Systems Software
Chapel Hill, NC
Level II, 16K
$12.95 on cassette

by Debra Marshall
***80* Staff**

I've had it! I'm sick of it! I hate it!... Well, maybe I'll try Deathmaze just one more time.

This has got to be the most infuriating, irritating, aggravating, frustrating, angering, spellbinding game on the market. Whoever wrote it is a pure nut.

I'm not going to describe it, except to say that it's different from any other adventure game I've seen so far. Just thinking about it makes my blood boil, my teeth chatter, and I get an overpowering urge to try it again... just once more. One staffer here actually dreams about the corridors and doorways.

The game comes complete with practically no directions whatsoever, but it does have a warning: "Be patient. You will not solve Deathmaze during the first week. Or the first month." Well, I've managed to involve just about everyone at Green, Inc. in this mind trap, either directly or indirectly. My boss is making dire predictions about my career, when he's not making suggestions about how to solve this puzzle. I'm getting desperate. What do I do? I can't get past the first level!

To Everything There is a Season

I've wandered the maze, checked out all the rooms, and gathered all sorts of junk. Somehow this stuff is supposed to help me get out of the maze alive. It isn't happening. What would you do with a rotten sneaker, a jewel handled dagger, a precision crafted frisbee, a hat with ram's horns, a broken calculator, and the boxes all of them come in? Or what I am supposed to make out of the cryptic message I found on one wall: "To everything there is a season"? I mean, nothing works!

I'll tell you what I keep doing. I keep starving to death. I've starved to death by individual and group effort, at least 50 times in the last two weeks. A person's got to eat somehow! I can't even get out of the seventh room to find the fridge.

Please, please, doesn't someone out there know where to find food and how to get out of the seventh room? Where's the door to the second level? Does anyone have a magic word to activate any of the junk I've been carrying around? Do I really want to put myself through five levels of this agony?

Ah, but such enticing, interesting agony!

I've got to get back to work now. I left some articles here on my desk somewhere... Help! Oh no. There's that crazy message again!

Invert and telephone.
Invert and telephone.
Invert and telephone.
Invert and telephone.

Maybe I should try.... ■

Note: In an attempt to preserve the sanity amongst the staff at 80 *and* Kilobaud, *and in order to get some real work out of these game-crazed editors,* 80 Microcomputing *hereby offers a reward of $50 to the first person to provide the solution to this game. Included will be our grateful thanks.*

The Patch
Cecdat, Inc.
Moscow, ID
$69.97

by Sal Navarro

Let me tell you of the events leading up to my selection, purchase and installation of the Patch.

I've installed many dual case mods for friends of mine, each one just a little bit different, but still basically the same. Finally, Radio Shack came out with theirs. Although it had a different graphics generator to give it descenders, and considering the fact that it won't work with all programs—it remains basically an Electric Pencil mod. All of the dual case mods I've encountered need to reserve memory for a driver routine, and some programs I have conflict with the dual case drivers. Then, along comes The Patch.

One day I was looking at an advanced copy of an advertisement for *Kilobaud* and *80 Microcomputing* on the Patch. The ad had statements like 'Electronic Module,' 'No software to load,' 'Operates like a standard typewriter,' 'Block cursor,' 'Special options available for five to ten dollars more.' It convinced me to look into what The Patch could do.

I called Cecdat and spoke to Greg Mattson, who told me that the first modules had been shipped and were being used with great satisfaction. After a lengthy talk and Greg's assurance that if there were any problems, (which there shouldn't be) they would be promptly taken care of. It was then I decided to order my Patch.

The Dawning of the Day

The day it arrived was truly not one of my better days. Murphy's laws were in full bloom. When I arrived home, on the table lay a package marked The Patch. The way the day had gone I wasn't sure if I wanted to open it. I found the instructions aimed at a novice user. I noticed that if you already had the Electric Pencil mod in, you could skip over two thirds of the instructions.

After supper and a little relaxation while reading over the instructions again, I decided to tempt Murphy. It went along flawlessly and in 30 minutes I was ready to power up. Lo and behold, on my video screen was upper and lowercase. Needless to say I had to try each and every letter from A to Z. First lowercase then uppercase. Everything was there, like a regular typewriter. Press shift and there was uppercase, no shift and I had lowercase and I didn't even load a driver routine.

I then proceeded to try TRSDOS 2.1, 2.2, 2.3, NEWDOS, and VTOS 3.0. BASIC will accept lowercase but DOS will not. Fortunately, I had some zaps for NEWDOS to allow DOS to accept lowercase commands. After I applied them, NEWDOS accepted commands with uppercase, lowercase and a mixture of them. CP/M was the only funny one. It accepted lowercase input but appeared on the screen in uppercase. Also to stop a BASIC listing the shift @ would not work. Upon further investigation I realized that it was looking for a reversed keyboard. When I tried '@' with no shift it worked fine. When you scroll a program it is much easier to use the @ without the shift to start and stop the scroll.

Next I tried my disk directory program because it resides in high memory and uses the rest of memory to record and manipulate the records. It worked perfectly! In highsight though, I see where the programmer was a little careless with the menu and some other statements. There are a few upper and lowercase letters mixed together which I had never seen before.

Murphy's Law #15

Next I entered free and was greeted by 'program not found.' What program? I asked for free space on the disk. Then I noticed that I had only typed 'fre.' Pardon me computer. Something was wrong, I couldn't get that second 'E'. No matter how many times I hit that 'E' key, it would not print. I tried a different key and it worked. Then I went to the 'E'; it worked. I tried it a second time. No good. Finally I found I had to wait about 15 seconds before the key would work again. What could Murphy be doing to me?

It turned out to be Murphy's law #15. One debounce in DOS plus one debounce in the Patch equals one key dead for 15 seconds. After I disabled the debounce in NEWDOS, everything worked fine. I continued to try different programs to see if there were any other strange problems that I might run into. There were none.

By now, with the time difference between New Jersey and Idaho, I could place a call to Cecdat. After I explained the problem to Greg we both had a good laugh and he said that if I wanted a permanent fix to send him back the Patch and he would remove the debounce. When I received the Patch back I tried a single key over and over again. Greg had removed the debounce as promised.

I Love My Patch

As you have probably figured out by now I don't particularly care for Murphy but I do love my Patch. I highly recommend The Patch to anyone who is interested in having a reliable dual case system with no software to worry about loading or losing.

I understand that Cecdat will be announcing another version of the Patch for $97.47. This new version will be able to toggle between a standard TRS-80 keyboard and an upper/lowercase keyboard. Also they will be introducing an electronic shift lock for $10.00.

An extra plus is the one-year warranty you get with the Patch. By the way, I received my Patch in February 1980 and it has worked ever since without a hitch. ■

80-Graphix Board
Programma International
Los Angeles, CA
$150

by Bruce Douglass

The 80-Graphix board is offered by Programma International as a solution to the TRS-80's low-resolution graphics. The 80-Graphix board, the ad says, gives you greater resolution than the Apple II. In return for high resolution you get a programmable graphics generator complete with lowercase.

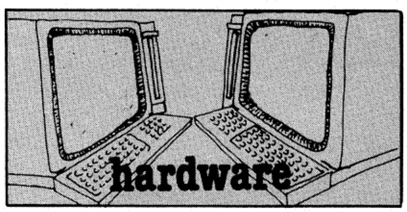
Ease of Installation

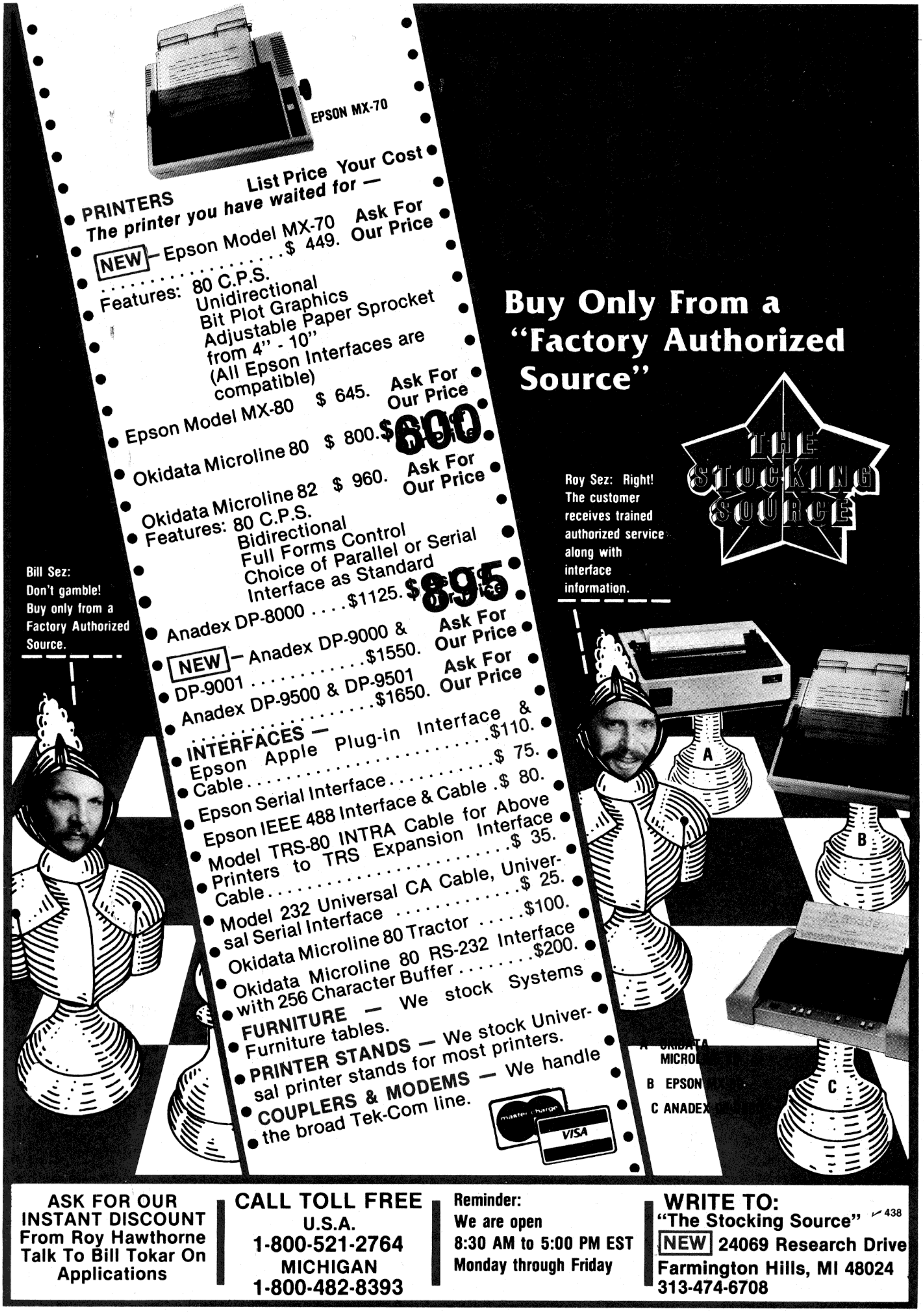
The board comes with easy-to-follow directions. I managed to install the board with only one slight modification in the directions. One of the ICs was "hunchbacked" and wouldn't stay in place for the cement to dry. I ended up soldering it into place.

Straightforward Programming

Programming the 80-Graphix Board is straightforward, but consumes a great deal of memory. The video memory is used to program the characters in the character generator.

The board offers three graphics modes: low resolution (normal graphics), high res-




EPSON MX-70
List Price Your Cost
PRINTERS
The printer you have waited for —
NEW – Epson Model MX-70 ....$ 449. Ask For Our Price
Features: 80 C.P.S.
Unidirectional
Bit Plot Graphics
Adjustable Paper Sprocket
from 4" - 10"
(All Epson Interfaces are compatible)
Epson Model MX-80 $ 645. Ask For Our Price
Okidata Microline 80 $ 800. $600
Okidata Microline 82 $ 960. Ask For Our Price
Features: 80 C.P.S.
Bidirectional
Full Forms Control
Choice of Parallel or Serial
Interface as Standard
Anadex DP-8000 ....$1125. $895
NEW – Anadex DP-9000 & DP-9001 ....$1550. Ask For Our Price
Anadex DP-9500 & DP-9501 ....$1650. Ask For Our Price
INTERFACES —
Epson Apple Plug-in Interface & Cable ....$110.
Epson Serial Interface..........$ 75.
Epson IEEE 488 Interface & Cable .$ 80.
Model TRS-80 INTRA Cable for Above Printers to TRS Expansion Interface Cable.......................$ 35.
Model 232 Universal CA Cable, Universal Serial Interface ....$ 25.
Okidata Microline 80 Tractor .....$100.
Okidata Microline 80 RS-232 Interface with 256 Character Buffer ....$200.
FURNITURE — We stock Systems Furniture tables.
PRINTER STANDS — We stock Universal printer stands for most printers.
COUPLERS & MODEMS — We handle the broad Tek-Com line.
master charge
VISA
Buy Only From a "Factory Authorized Source"
THE STOCKING SOURCE
Roy Sez: Right! The customer receives trained authorized service along with interface information.
Bill Sez: Don't gamble! Buy only from a Factory Authorized Source.
A
B
C
B EPSON
C ANADEX
ASK FOR OUR INSTANT DISCOUNT From Roy Hawthorne Talk To Bill Tokar On Applications
CALL TOLL FREE
U.S.A.
1-800-521-2764
MICHIGAN
1-800-482-8393
Reminder:
We are open
8:30 AM to 5:00 PM EST
Monday through Friday
WRITE TO:
"The Stocking Source" 438
NEW 24069 Research Drive
Farmington Hills, MI 48024
313-474-6708

How's your love life?

A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!

Interlude is: romantic... playful... outrageous... a fantasy. Interlude is: ■ A Bed of Roses (Interlude #1) ■ Mata Hari (Interlude #49) ■ The Chase (Interlude #7) ■ Rodeo! (Interlude #71) ■ The King and I (Interlude #60) ■ Some Enchanted Evening (Interlude #84) ■ Caveman Caper (Interlude #82) ■ From Here to Ecstasy (Interlude No. 30) ■ Satin Dreams (Interlude #72).

More than 100 Interludes are included in the program. Most are described in detail in the accompanying manual, but several surprise Interludes are buried in the program awaiting that very special time when your interview says you're ready. (When you learn secret Interlude #99, your love life may never again be the same!) Interlude can give you experiences you'll never forget. Are you ready for it?

Interlude™
The Ultimate Experience. ~287

INTERLUDE, 10428 Westpark, Houston, Texas 77042. I'm really ready. Send my Interlude today.

Apple II (16K)*
☐ Cassette ($16.95)
☐ Diskette ($19.95)
☐ Diskette—Pascal or DOS 3.3 ($19.95)
Add $1.50 for shipping and handling.

TRS-80 (Level II-16K)**
☐ Cassette ($16.95)
☐ Diskette ($19.95)

Poster
☐ 20″x 24″ reproduction of this ad without ad copy ($4.95—includes shipping charges)

Available for immediate shipment.

Please enclose your check payable to INTERLUDE or complete the charge information:

☐ MASTERCARD ☐ VISA All charge customers must sign here.____________
Account No. ____________ Expiration date ________ MasterCard Bank Code ________

CHARGE CUSTOMERS: Order by phone toll-free! **1-800-231-5768 Ext. 306** (Texas: 1-800-392-2348 Ext. 306)
Name ____________ Age ______
Address ____________
City ____________ State ________ Zip ________

*Apple II is a registered trademark of Apple Computers, Inc. **TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

olution, and program modes. These modes are accessed through the cassette port, which has one major advantage: They can be easily accessed in either BASIC or assembly language.

In BASIC, OUT 255,32 puts you into normal graphics; while OUT 255,160 puts you into the high resolution mode; and OUT 255,96 puts you into the program mode. So far, so good.

Programma says that you can use their graphics and still access your normal graphics. True enough, but that doesn't mean you can do both at the same time. You are either in the high resolution mode, the program mode or in normal TRS-80 graphics mode.

Programma also says that you get lowercase. Well, that sounds like a pretty good deal, but what you get are graphics characters that only look like lowercase ASCII.

When you send these letters to a printer, your printer reverses upper and lowercase: That is, uppercase as displayed on the screen is printed as lowercase by the printer and vice versa. Since this has to do with the lowercase print driver itself, it can be modified to correct the transposition.

Once you are in the program mode, the screen is blanked out, and your previously devised data statements POKE the desired characters into the board's RAM.

The 80 Graphix characters are composed of a rectangle of 6×12 pixels. In program mode, you have access to all 72 individual pixels in each byte, and you can turn on each individual pixel as desired. These pixels are the same ones that make up the ASCII characters on the CRT display. If you look at them (ASCII) you will get an idea of the resolution we are talking about.

Normal TRS-80 graphics characters are composed of a rectangle of 2×3 pixels. These pixels have the added advantage, however, that they can be accessed individually in your display by using SET and RESET.

To SET a pixel, its position must be XORed with the graphics character in that byte of video memory. The characters are arranged in such a way that the new graphics character will have that pixel SET, as well as the other pixels that were previously SET. Note that this is done by putting a *new* character into that byte of video memory.

Now that leads us to the point (finally). Unless you are extremely clever (much more so than I), you will be totally unable to SET and RESET while in the high-res mode with the 80 Graphix Board. The Graphix Board will display the character whose CHR$ number is the XOR of the sixth-bit pixel position and the character currently in memory. This character, however, will have nothing to do with the normal meaning of SET and RESET.

With only byte-resolution and display, and without the help of SET and RESET, and with only 64 characters available to program, it seems to be a misnomer for Programma to state that you have 384×192 resolution. 64×6×12 makes more sense.

The memory locations starting with the video memory (15360 in decimal or 3C00 in hex) are used to store the bytes that define your character set. Each row of six pixels is coded in one byte. Therefore, as you might expect, it takes 12 bytes to code each character. The next four bytes are spacer bytes and are ignored. To code for all 64 characters requires the entire video memory while you are in the program mode.

To set every other bit, the number (in binary) is 010101. You take this number and multiply it by two (or shift it left in assembly language) and set bit seven (add 128 decimal). This is the number that you will POKE into the first byte of video memory.

If you wish to save some memory, you don't have to code the four spacer bytes between each character set. Still, generating characters takes up an entire 1K of memory. Further, while generating characters, you cannot use the CRT because nothing is displayed. Once you leave the program mode and enter high resolution mode, your screen returns.

Advertising Claims

As I previously mentioned, the advertisement that Programma International runs is not entirely as it sounds.

The ad states that several demo programs are included with the board, besides software to aid in the development of graphics characters. Several in this case means three: a fairly long one that extols the virtues of the Programma name and the 80-Graphix board, a CRT version of a Star Wars spacecraft (which look good) and the aforementioned lowercase generator.

Their ad shows a powerful looking character generating program and what you get, "Create," looks somewhat similar, but is less useful. The software shown in the ad displays the entire character set, and says which character you are on.

The program you actually get only works with one character at a time, and does not display or construct an entire set, as does the one in the ad. This makes it harder to visualize how your characters are supposed to fit together, and in fact, whether or not they actually look like you want them to.

Now to use this generator in a programming situation, you must first define all your characters, one byte at a time, write down the data statements and type them into your program.

There is one last point I would like to make to those of you searching for high resolution graphics on the TRS-80. You must assess your needs before you buy. If what you really want is vector graphics, then I can not recommend this board.

However, if a graphics generator is your heart's desire, and you are willing to spend a rather large amount of time programming the board, the 80-Graphix Board may be the ideal add-on for you. Once programmed, the switch between high resolution and normal graphics is easy in both assembly code and in BASIC.■

Line Printer III
Tandy/Radio Shack
Ft. Worth, TX
$1999

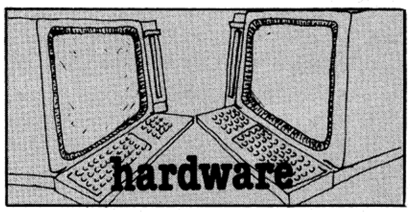
by Robert James Lloyd

Have you ever wanted a hard copy listing of a special program? Or tried to debug a 250-line program while only looking at the video display?

If any of the above were answered in the affirmative, then you have a very real need for some type of printer. Such was my problem. Soon after I became a microcomputer hobbyist, I began the search for a printer that met my needs.

A friend of mine was able to attend the New York National Computer Conference last year, and as a favor, brought back some of the literature distributed by the many exhibitors. Needless to say, I was engrossed for several hours, during which time I digested the pros and cons of various printers available for the TRS-80. Graphics, friction feed, tractor feed, upper/lower case; the list goes on and on.

Being the contented owner of two Ra-



dio Shack computers, I decided to continue with their products. I wanted a printer that could handle numerous forms without locking me into any one type of paper such as thermal or roll stock. While I have nothing against these printers, I wanted to be able to print labels, envelopes, and different kinds of multi-part forms. All the requirements I desired were found in Radio Shack's Line Printer III.

Retail price of Line Printer III is $1999! Admittedly, two grand is a rather steep price for the home hobbyist, but consider its features:

- 120 cps/50 lpm.
- Upper/lower case.
- Bi-directional, logic-seeking carriage.
- ASCII characters from decimal 32 thru decimal 127.
- Expanded character capability, software selectable.
- Handles forms up to 15 inches in width and up to five parts.
- Adjustable tractor feed for alignment accuracy.
- 132 characters per line (64 with expanded character format.
- Compatible with Model I or II.
- Can be used with or without expansion interface. (Requires Printer Interface Cable 26-1411 for operation without EI).
- ONLINE/OFFLINE modes.
- Paper out sensor stops printer when end-of-forms is reached.
- Six or eight lines per inch, software selectable.
- Manual controls for line feed.
- Rear or underneath paper feed.
- Manual self-test.
- Ribbon in cassette form to allow easy replacement.
- 9×7 dot matrix print head.

The one feature lacking is graphics. Perhaps, one day, Radio Shack will market a graphics modification for the printer as they did the lower case mod for the computer. (Are you listening, RS?)

Several other negative characteristics came to light during my first week's ownership. Most are built-in quirks that take getting used to, but one is just plain economics. Below are the quirks:

- Line Printer III automatically prints 132 characters per line unless a carriage return is received. When listing programs, line statements should not be longer than 8090 characters for 8 1/2" + 11" paper. Longer lines will require 15" forms.
- You cannot backspace the printhead or overstrike characters (such as when you wish to underline).
- The first form fed in is wasted because of the print area's design. This is especially annoying when printing checks or using pre-numbered forms.
- On/Off switch is located in the rear. This is not an actual deficiency, just my personal preference. I like all controls and switches in one place.

Now the economics. Two thousand dollars for the printer was high, but I accepted it. I realize inflation is taking its toll. But I can't believe the price of a replacement ribbon: $21.95! I have yet to find any at a cheaper price. Mine only last approximately three to four months which amounts to $65-$90 annually. And if you use Line Printer III in a business, ribbon costs could exceed several hundred dollars a year!

Set-up can be accomplished in 15 to 30 minutes, depending on how anxious you are. Follow the operating instruction booklet and you should have no trouble. If you are using the Printer Interface cable, follow instructions outlined in the booklet for proper connecting procedures.

Because the instructions are clear and straightforward, there's no need to go into the procedure for conducting a self-test. Once this has been accomplished, you should be ready for a trial run. My self-test consisted of a simple program listing. At no time did I experience any problems or malfunctions. Line Printer III has worked each and every time it has been powered up without error.

There are several hints I feel that I should pass along to help the beginner. If you should print in the expanded character mode, be sure to reset it back to normal mode. This must be done via a software command. Otherwise, the printer will produce expanded characters until it is powered off and on again. The same applies when printing eight lines per inch.

Line Printer III weighs in at a hefty 50 plus pounds. For this reason, Radio Shack recommends that a printer stand be used. I didn't have the funds available to buy one after purchasing the printer, so I made one utilizing 1/2" plywood and several two-by-fours. This allows feeding forms directly from underneath.

For you electronic wizards, diagrams of the controller logic, power supply logic, driver logic, and motherboard logic circuits have been included in the instruction booklet. I would, however, caution anyone from trying to make repairs on their own. If repairs by Radio Shack become necessary, their policy is to charge a higher rate if there is evidence of tampering. Also you would automatically void the warranty.

Radio Shack has a great printer here. Positives far outweigh any negatives. I have had my Line Printer III for almost a year and have come to appreciate the luxury of owning it. I would not hesitate to recommend the purchase of a Line Printer III to anyone. The one major advantage, I feel, is the readily available service—without having to re-package and pay shipping and insurance—when and if repairs become necessary.

Should you decide to buy, you will be getting a valuable addition to your computer center. With proper care and maintenance, you will experience trouble-free operation. Happy printing! ■

32		33	!	34	"	35	#	36	$
38	&	39	'	40	(	41	)	42	*
44	,	45	-	46	.	47	/	48	0
50	2	51	3	52	4	53	5	54	6

Fig. 1. Line Printer III Character Set—ASCII Codes in Decimal

32		33	!	34	"	35	#
38	&	39	'	40	(	41	)
44	,	45	-	46	.	47	/
50	2	51	3	52	4	53	5

Fig. 2. Expanded Character Set

META TECHNOLOGIES

FOR YOUR TRS-80™ DISK SYSTEM

PROGAMMING TOOLS

NEW LOW PRICE!

TDAM $14.95
For Model II $24.95
Includes MTC QUE Card!

Having trouble with RANDOM FILES? With MTC's Table-Driven Access Method (TDAM) you'll never fret over FIELDing again. No knowledge of random access files is required. Insert the TDAM "interpreter" into any BASIC program and type in a few DATA statements describing the information in your files. TDAM does the rest! Reads and writes fields and records of any type (even compresses a DATE field into 3 bytes!). Features automatic file buffer allocation/deallocation, memory buffering, sub-record blocking/deblocking, and handles up to 255 fields per record. Super fast and super simple! Complete with TDAM interpreter, instructions and demo program. Requires programming experience.

SIFTER $14.95
For Model II $24.95

Twelve in-memory high-speed sorts for use in any BASIC program: stable, non-stable, with/without tags, for numeric or string data. Random File Sort included. Some sorts written in machine code. Includes sort subroutines, demo programs and instructions. Relocate as needed with REBUILD. Requires programming experience.

SHRINK $14.95
For Model II $24.95

Makes Every Byte Count! Make programs smaller and faster! Combines lines & removes unnecessary code including remarks, without altering program operation. Typically reduces program size 25% to 40%.

SUPERSEDE $14.95
For Model II $24.95

A "must have" for the professional programmer or the serious amateur. Probably one of the greatest time-savers available. Write programs in shorthand - change variable names - generate program documentation - use with REBUILD and MINGLE to build new programs from old ones.

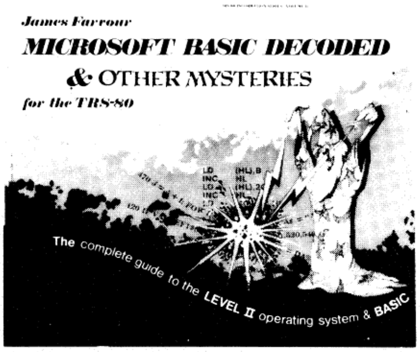
MINGLE-II $14.95
For Model II $24.95

Merge up to 14 files (Program or Data) into a single file. Data files may be merged in ascending or descending sequence with the ordering based on a user-specified comparison field. A very handy utility for consolidating data files.

"OTHER MYSTERIES" VOLUME II

foreword by

H.C. PENNINGTON



Call now and place your order for this new book, "MICROSOFT™ BASIC DECODED & OTHER MYSTERIES for the TRS-80™", from IJG, Inc. A primer for cassette and disk BASIC on the TRS-80™, the information provided applies to similar MICROSOFT™ BASIC interpreters. Features include definition of terms, an overview of BASIC and DOS, explanation of exits, error codes, verb actions, "cold" and "warm" restart procedures and examination of system utilities, arithmetic support and I/O driver routines, and the communications region in RAM. Individual routines are explained in detail, with an index provided for easy access. Appendixes include tables for BASIC and DOS vectors, stacks and interrupt locations, PLUS thousands of comment lines for the complete MICROSOFT™ BASIC.

MICROSOFT™ BASIC DECODED .. $29.95

The perfect supplement for your NEWDOS, from IJG, Inc.

"TRS-80™ DISK AND OTHER MYSTERIES"

by Harvard C. Pennington

132 pages written in PLAIN ENGLISH packed with HOW TO information with details, examples and in-depth explanations. Recover lost files and directories, remove file protection, make BASIC programs unlistable. How to use SUPERZAP, recover from DOS errors and MORE!

TRS-80™ DISK $19.95

NEWDOS/80

by Apparat

Apparat's long-awaited successor to NEWDOS+ is here! This is not an enhanced version of NEWDOS, but a completely new product. Simplified DOS commands can be instantly executed from BASIC, even within a program, without disturbing the resident code. System options, such as password protection, number and type of disk drives, BREAK key enable/disable and lowercase modification recognition, can be quickly and easily changed. Five new random-access file types allow record lengths of up to 4096 bytes, and no FIELDing! A powerful CHAIN facility allows keyboard INPUTs to be read from a disk file. An improved RENUMBER facility permits groups of statements to be relocated within program code. Diskettes may even be designated as RUN-ONLY! Features all NEWDOS+ utilities (SUPERZAP 3.0, etc.) and much more! One MTC technical staff member said having NEWDOS/80 is "better than sex" (you'll have to judge for yourself!). Includes 180-page instruction manual and MTC QUE card.

NEWDOS/80 .... $ 149.95
MTC OUE Card only $7.50

CALL REGARDING OUR NEWDOS + UPGRADE PRICING

Complete for Model I with all utilities Plus exclusive MTC QUE card!

NEWDOS +

$69.95 by Apparat

includes REF, RENUM, SUPERZAP, EDITOR/ASSEM., DISASSEM., DIRCHECK, and more! This is the original NEWDOS with all of Apparat's utility programs. Includes exclusive MTC QUE (Quick User Education) card.

40-Track Version $79.95
MTC QUE Card only $1.50

MORE PRODUCTS

Products damaged in transit will be exchanged. Prices, Specifications, and Offerings subject to change without notice.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

DEALER INQUIRIES INVITED

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $2.50 for standard UPS shipping & handling
- $2.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

FOR PRODUCT INFO 1-800-321-3640

CALL TOLL FREE

TO PLACE ORDER 1-800-321-3552

IN OHIO call (216)289-7500 (COLLECT)

META TECHNOLOGIES CORPORATION ✓20

26111 Brush Avenue. Euclid. Ohio 44132

801219

TRS-80 is a TM of Tandy Corp.
©1980 by Metatechnologies Corporation, Inc.

EDUCATION 80

by Earl R. Savage

"Nothing but computer language is written in uppercase. If you are going to teach people to read, you must have lowercase on that screen."

By this time you're surely aware that Radio Shack made several goofs in the design of your 80. If you didn't discover them yourself or read about them, I'm sure your students told you about them. I used to hear a lot of gripes, but a number of the problems were solved by RS. They put those solutions in later production runs and even retrofit older machines at no cost.

There is one problem which they have not corrected. For want of one cheap integrated circuit chip and a few wires, you are limited to uppercase letters on your screen. You know the lowercase letters are in there somewhere because they come out on your printer. Somewhere on the way to the display, everything is converted to caps.

Critical Shortcoming

The lack of lowercase display is of little if any consequence in most applications, but in several educational uses this shortcoming is critical.

Have you tried using 80s for typing instruction? Sure, it *can* be done with the display all in uppercase, but it cannot be done well. Looking at caps while typing small letters can create the wrong motor responses in the learner. Frankly, to me, this is unnecessary.

And how about reading? It doesn't matter whether we speak of teaching speed reading or remedial reading or reading to the little tykes in the primary grades—the problem is the same. Nothing but computer language is written in uppercase. If you are going to teach people to read, you must have lowercase on that screen.

For a variety of reasons your 80s would be more useful if they displayed lowercase letters. RS *will* install a lowercase modification for you, for a fee. (Note the lack of an "r" in that last word.)

On the other hand, if the powers-that-be in your school will permit it, you can make the modification right in your room for the buck or so the IC costs. If you can't do it yourself, round up help from the electronics or physics teacher, an advanced technical student, or even a cooperative parent.

There have also been many magazine articles containing directions for adding lowercase. My own preference is found in the April 1980 issue of *Kilobaud Microcomputing.* It is simpler than most and works like a charm. Finally, there are a number of lowercase kits available for around $20 (still much less than Radio Shack's modification fee).

When I installed my first lowercase mod, I was surprised to find that some of the small letters looked strange. The letters with descenders (p,q,g,etc.) sit up on the line instead of descending below it. Those letters are clearly different from their uppercase counterparts, and you will find it takes a while to get used to having them there.

The practical way to real descenders is to replace the IC character generator. I understand that all but the earliest 80s have that chip in a socket, making its replacement quite easy. The only source I have found for a "proper" generator IC is EBG & Associates, 203 North Wabash, Chicago, IL 60601. No wiring changes are required with this IC, you just pop out the old and pop in the new . The generator costs about $20 and it makes a beautiful display.

> *"The letters with descenders . . . sit up on the line instead of descending below it."*

Speed Mod

I suggest that you make another modification while you are in the 80, to speed up its operation. The 80 usually operates with plenty of speed, spending most its time waiting for the student to respond or input data. There are times, however, when the student sits and waits for the computer to complete a function (CLOADing, CSAVEing, searching, sorting, etc.)

Radio Shack will not modify your 80 for faster operation, but you can do it. There are speed-up kits available—again, for that magic sum of $20. For do-it-yourselfers, the February 1980 issue of *80 Microcomputing* explained one method.

Do not be disdainful of the 50 percent increase in clock speed that most modifications provide. Fifty percent faster is significant in all machine operations, yet not so fast as to be unreliable.

We have been using the homebrew lowercase and speed modifications described above; each worked on the first try and neither has given any trouble in countless hours of operation. We have found them to be a great advantage.

Incidentally, you can turn these particular mods on or off with keyboard commands or with statements built into your programs.

Perhaps I had better be completely honest about that "no trouble" statement a few lines back. The speed mod did cause some hard times when first installed. You simply can't CLOAD at one speed, if a tape was CSAVEd at a different one. We made several unsuccessful attempts at first, and suspected the 80 or the recorder or the cassette or the cables or all of them before it finally dawned on us—change the speed.

No-Repeat Selection

There you sit, writing an instructional program, which randomly selects questions or problems from a list. Or perhaps you are using the random function to generate an endless series of math practice problems. How can you prevent a question from being repeated on the very next item?

Take a look at this procedure:

```
120 Y = X
130 X = RND(30)
140 IF X = Y THEN 130
150 PRINT A$(X)
```

In line 120, the last completed item identification is filed in Y and the new selection is made in line 130. They are compared in line 140; if they are *not* the same, execution falls through to line 150, which causes the new item to be displayed.

Although two consecutive items cannot be identical, there can be repetitions as long as there is at least one other item between them. If you wish to prevent the repetition of the last two questions, make these changes:

```
120 Z = Y:Y = X
140 IF X = Y OR X = Z THEN 130
```

This procedure assumes that your questions are stored in array A$(n). It is easily modified to suit other types of question storage or generation, and can be expanded to prevent duplication of as many previous items as you wish.

There is a better approach to non-repetition when the number of items to be avoided becomes large. Let's look at the case of a series of random questions about the 50 states when you want no repetitons. We'll put the items (states) in array A$(n). These statements would be placed in the program:

```
90 M = 0
100 M = M + 1
110 IF M = 51 THEN 170
120 X = RND(50)
130 IF T(X) THEN 120
140 T(X) = 1
150 PRINT A$(X)
160 GOTP 100
```

This procedure sets up numerical array T(n) to test for repetition. Each T(n) is initially equal to zero. When X is selected in line 120, line 130 tests to see if T(X) equals zero. If not, line 120 selects X again. If T(X) does equal zero, execution falls through to line 140.

In line 140, T(X) is changed from zero; that is, the flag is set. This means that the same value of X cannot get past line 130 for the remainder of the RUN and the same state will not be printed again. Lines 90, 100, 110, and 160 assure that 50 and only 50 items are selected.

Either method for preventing repetition of randomly selected items can be used with discrete items in an array or with items that are generated.

Help For Other Readers

We all run into problems we must work out or work around as we write programs. Sometimes the solution can be complex. Often, however, a short subroutine or a few program statements will provide a solution.

You must have worked out a few problem solvers yourself. Wouldn't you like to share them with other readers? Send them to me with a brief explanation of the function of each. I'll put those of greatest interest in future ED 80 columns as space permits. Of course, you will be listed as the contributor. ■

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

It seems that no matter how much effort and care you put into learning a new piece of equipment, old Murphy is not far away. Despite all the time that I spend at this gray box, I still have much to learn.

This is a classic example of the danger of violating one of the most basic principles of microcomputer application. *Never mix equipment from different manufacturers!*

One of my clients recently installed a Model II system. Because they already owned a Teletype Model 40 printer they asked me if it could be connected to the Model II.

I reviewed the Model 40's specifications and found that it could be equipped with an EIA-compatible serial interface. The Model II is equipped with two EIA-compatible serial ports, and a serial printer driver is available which can be invoked by using the FORMS "S" command.

There appeared to be no problems in interfacing the two devices, and I advised my client that the hookup appeared feasible. The client then contacted Teletype service to install the proper connectors and ordered an interconnecting cable from Radio Shack.

Naturally, the first try at a hookup was a failure and I was asked for assistance. I quickly found several inconsistencies. The Teletype required two more connections than the computer provided. After much head scratching, the Teletype technicians finally succeeded in developing an interconnecting scheme that should have worked. Unfortunately, the Model II refused to recognize the printer.

At the client's request, Teletype sent over one of their top technicians with some special equipment. It quickly became apparent that the computer was not sending data to the printer. I switched to the CP/M operating system. With this system and MBASIC 5.0, I could read the status of the serial ports and send data directly to the printer using BASIC INP and OUT commands. These commands aren't available under TRSDOS. MBASIC let me send data to the printer; however the printer did not recognize carriage returns or line feeds without a delay loop to insert nulls.

More Disturbing

While this was only inconvenient, a more disturbing factor appeared. When printing was interrupted for more than 40 seconds, the printer turned itself off. The only way to restart it was to reboot the system and re-initialize the interface. You could restart the printer without resetting the computer if the motor start pin was briefly pulsed. Unfortunately the logic of the printer did not allow hardwiring the motor start pin to a permanently-on position. This had been done deliberately to avoid burning out the printer motor.

The problem, then, was the software driver.

Fortunately, I had had experience with custom printer drivers in the past. I knew that the development of driver software for a Model II TRSDOS system was a complicated undertaking.

Radio Shack does not document driver software. When an assembly language programmer wishes to use a peripheral he uses service-call routines. The service-call routine uses vector interrupts and a jump table to locate the desired routine. If the driver program is suitable, all goes well. If not, altering the driver is *rather* inconvenient. First, the driver has to be found to be modified.

I just ran into these problems when I purchased a surplus Selectric typewriter. Although the driver program and conversion table was supplied with the Selectric, it took almost a month to get it working with TRSDOS. Based on this experience, I estimated that an assembly language program could require two man-weeks to develop and interface with TRSDOS. At an estimated $70 per man-hour, such a program could cost approximately $4,900.

Faced with this information, my client took the easy way out and ordered a Line Printer IV. He chided me for suggesting that the Teletype would work.

By now I'm sure you are curious what happened to the Selectric: I sold it. As I had guessed, Radio Shack modified all the drivers when they issued Scripsit II on TRSDOS 2.0, and the Selectric no longer worked. Facing another month of interfacing problems, I gave up.

Taxes

As I write this column, we are deep into our end-of-year tax planning procedure. It is this planning procedure that a professional uses to help clients make proper decisions to minimize taxable income.

To aid in the tax projection procedure, many accountants use worksheets that summarize tax information in various categories. Typically, these categories are quite broad with little detail, such as dividend income, capital gains, and partnership distribution. But these are professional procedures.

A program for the average taxpayer is Micromatic Programming Co.'s Tax/-Saver. Although the product we reviewed was not in its final form, it is an excellent example of conversational software. In this package, the computer becomes the accountant and interviews the taxpayer.

Realizing that this tax preparation program was not designed for a professional, I asked my good friend and non-accountant, Gordon E. Lamb, to look it over and give me his opinion. Here in his own words are his comments.

. . . .

A Review

PLEASE TYPE IN THE NUMBER OF THE SITUATION WHICH DESCRIBES YOUR (LAST) *MARITAL STATUS* ON DECEMBER 31 1979. Thus Tax/Saver begins an in-depth interview of the taxpayer. As its conclusion, even the neophyte, unfamiliar with either IRS or TRS, will have been (in most cases) successfully coached through his or her long or short 1040 form.

As supplied in the preliminary review copy, the program consisted of three unformatted disks and documentation. It demanded the services of a two-drive Model I. The version I reviewed did not support a printer and required that you copy information from the screen to your tax forms. The system provided no way to stop in the middle of a long return, but rather demanded that you finish the work at one interrupted sitting, free from power outages or voltage surges.

After a bit of beginning graphics, the program embarks upon a well thought-out interrogation of the user. Initially it goes about the business of creating a generalized picture of the taxpayer. It offers advice as to whether to file the short or long form, compares the merits of married couples filing jointly or singly, and recommends data to have on hand before beginning the return.

All information is gathered in such a way to allow easy revision after entry. To maintain accuracy, the questions double-check your entries. As an example of the program's detail, consider the selection of suitable dependents. A brief description of qualifiying characteristics is displayed followed by a query as to whether you need any assistance in screening a particular individual. The program then offers a series of tests which will either qualify Uncle Fred, disqualify him outright, or remind you that you may, for example, claim his medical expenses, even though you may not list him as a dependent.

At various stages of the operation, you are presented with a very complete menu of deductions and options, each introduced with a general description and remarks as to suitability. You are also advised as to the specific documentation needed to support each option.

Tax/Saver even removes the need for a calculator, offering to total interest, dividends, etc. As a deduction is worked out, the user is presented with a national average claimed by other taxpayers in his or her income bracket. An option is offered to recheck the calculation of the deduction. Various irregularities, such as overpayment of FICA taxes, are screened out and information necessary to recapture them is provided.

Slides Smoothly

Although the final product offered for sale may lack some of the features of the preliminary copy, this program can really aid the layman. There is a gray area between the point where a non-accountant can prepare his own returns or should seek professional help. This program is certainly equal to many store-front tax preparation services and *could* be more sophisticated. It slides smoothly through adjustments to investment credit tax on items prematurely sold, alternate minimum tax, income averaging, etc.

Because of the mass of detail, regulations and figures pertinent to a tax year, changes in the IRS regulations can quickly make the package obsolete. The buyer should be prepared to subscribe to an annual updating service. Still, at $49 for two diskettes, the tab for tax work by the Tax/Saver (deductable) is not too bad. Tax/Saver is available from the Micromatic Programming Company, Georgetown, CT.

. . . .

After Gordon's well chosen comments, I can only caution that programs such as Tax/Saver may not be desirable in all cases. Tax law is complex, and no computer—regardless of how cleverly programmed—can equal a competent professional. For example, Tax/Saver does not handle Schedule D transactions. These transactions occur when capital assets are sold or exchanged. Despite these objections, Tax/Saver may very well live up to its name in your situation. It's worth a look.

This column marks the first time that I am using an outsider to assist me in reviewing a piece of software. I welcome comments from any Model I or II user who would like to share his experience. I would especially like to hear from users of specialized software such as real estate packages, cash register emulators, survey systems, or critical path scheduling systems. I believe that your comments could be of value to all our readers. Please do not hesitate to write me. Once again the address is: 42 Bulaire Rd., East Rockaway, NY 11518.■

Index Welcomed

While thumbing through my stack of *80 Microcomputing* magazines the other day, frantically searching for an article that I remembered seeing, I happened upon Mr. Klungle's Magazine Index program in the April 1980 issue.

My search ended right there. I didn't find the article I was originally looking for, but I did find a useful piece of software that will undoubtedly save me considerable time and frustration in the future. It is a most clever cure for the "now-where-did-I-see-that-article?" problem. Also, it serves as one more justification for buying a microcomputer.

One slight modification is necessary, however, when storing data on a tape that has a leader. Line 805 (as modified) is as follows:

```
805 CLS:E = 1:PRINT@130,"STORING "N$;" DATA ON
    TAPE":PRINT:OUT255,4:FORI = 1TO5000:NEXTI:
    PRINT#-1,N$.
```

This turns on the tape transport and advances the tape well beyond the leader, preventing an attempt to write on the leader.

By the way, as I was feeding data into my Magazine Index program, I located the article that I had vainly searched for earlier.

Gordon Hogue
Panorama City, CA

Tape Scripsit Mods

In the October issue, R. J. Lighton explained how to modify Radio Shack's Scripsit to make the line feed commands work properly with a Selectric-type printer. His instructions were for the disk version of Scripsit.

For the tape version, the memory locations to be modified are different: locations 60E0H and 60F4H must be changed from 0AH (line feed) to 0DH (carriage return). The obvious way to do this is with T-BUG, but unfortunately T-BUG occupies memory locations 4380H to 4980H, thus sitting right in the middle of Scripsit, which occupies locations 4300H to 69C5H.

This is the place to make use of the program by Irwin Rappaport (*80 Microcomputing*, January 1980, page 118) to move T-BUG to locations 7380H–7980H. First, move T-BUG. Then get back into command mode with #J 0072 or #J 1A19. From command mode, type SYSTEM and load SCRIPS. When the tape has loaded, *don't* answer the *? question with the usual /ENTER; instead jump back to T-BUG with /29568. Use T-BUG to change the locations listed above, and then to punch a tape of the revised SCRIPS (#P 4300 69C5 4300 SCRIPS). This tape can be loaded and used in place of the original SCRIPS, and the line spacings will work as they should.

C. D. Graham, Jr.
Ardmore, PA

THE ASSEMBLY LINE

by William Barden, Jr.

". . . a computer gets an interrupt for such things as running out of bottles on an assembly line, . . . and jammed pickle slicers, not to mention China Syndrome meltdowns."

This month we'll be talking about . . . excuse me, I hear the doorbell . . . "Hi, John, I'll . . . excuse me, I hear the phone . . . "Hi, Ron, I'll call you back, I'm writing the Assembly Line Column—interrupts.

If you can make sense out of the first paragraph, you know how interrupts work, in a nutshell. We'll expand on this introduction by talking about what they are, how they're performed in the Z-80 and TRS-80, how to build a simple trace program using the real-time clock interrupt, and the possibilities of doing foreground/background processing with the TRS-80. This last topic is especially intriguing as it makes possible such things as using your TRS-80 to monitor home burglar alarms and sense temperature while you simultaneously diddle around in BASIC!

Why Have Interrupts At All?

A good question. There are two basic reasons to have an external signal that the computer recognizes while it is in the middle of a program. The first of these is that someone might pull the plug—a catastrophe! The second is that a slow-poke device, mechanical or human, is finally ready with another piece of data to enter.

The first *raison d'etre* is obvious. It's nice and often essential to have a signal that indicates a catastrophic condition. Larger computers have power-fail interrupts that indicate that ac power is failing. In the space of one power-line cycle, status can be saved in non-volatile memory so that the bits and pieces can be reconstructed later.

Such catastrophes, of course, can be further qualified so that a computer gets an interrupt for such things as running out of bottles on an assembly line, empty prune juice dispensers, and jammed pickle slicers, not to mention China Syndrome meltdowns.

The second reason covers a wide range of conditions. If a computer is waiting for a very slow device, especially one that is not predictable (or asynchronous), then it's handy to have a signal that indicates when the device has data available. It's also nice to have the same type of signal to indicate when a more intelligent device has finished with the data.

If a computer can continue processing while an input/output device is waiting for the next key to be pressed, or the current line to be printed, or the next communication character to come in, then the overall efficiency of the computer system will increase greatly. The computer system can overlap processing and input/output operations, recognizing an I/O interrupt only when the next piece of data is available.

This concept is widely used in large computer systems, where many jobs are run concurrently. Job number one is run until I/O is called for. The I/O is started, and job number two is run until it requests I/O, and so on.

Many real-time systems are also interrupt-driven, handling some "background" task until a higher-priority "foreground" task interrupts the processing. In general, background refers to a low-priority processing function while foreground refers to a high-priority task that must be handled in real-time or near real-time.

Depending upon the computer system, there may be from two or three to hundreds of separate interrupts possible, sometimes with distinct levels, or priority groupings.

How Do They Work?

The normal interrupt procedure goes something like this: Somewhere in the CPU is a flag that says that interrupts are enabled or disabled. Unless this flag is set, an external interrupt will not cause any action. Depending upon the machine, the flag may or may not be remembered.

If the interrupt enable is set, an interrupt comes into the CPU as a signal on a system bus line (a high going to a low, for example). The CPU acknowledges the interrupt by sending an interrupt acknowledge signal. At this point, the requesting device may send over additional identification data, which will vector the interrupt to a special interrupt processing routine. This interrupt processing routine may be anywhere from several to hundreds of instructions long. There may be a separate interrupt processing routine for each interrupt, or one general-purpose interrupt handler for many interrupts.

When the interrupt handler is entered, the first thing it must do is save the environment, or the set of conditions that existed at the interrupt—the contents of CPU registers, the state of the flags, the location of the interrupted instruction, and so forth.

The address of the interrupted instruction is (usually) saved in the stack automatically, or in a predefined memory location. There is no way of predicting where the interrupt occured—it could have been directly after an instruction that has set CPU flags, or in the middle of a block move. Very few instructions are non-interruptible—mostly instructions concerned with the interrupt, such as setting interrupt enable and returning from the interrupt.

If the interrupt processing routine is going to use CPU registers, then it must save all of the registers it will use in the stack. Chances are some of its instructions will affect the flags, so the flags almost always will have to be saved. The flags and registers will be restored immediately before the return from interrupt.

TRS-80 Interrupt Structure

Like assembly language instructions, the interrupt structure of the TRS-80 is intimately tied to the structure in the Z-80 microprocessor. The interrupts available for the Z-80 *are* the interrupts for the TRS-80.

There are four sets of interrupts for the Z-80, but only two are used in the TRS-80.

The first of these is the non-maskable interrupt, or NMI. The NMI is normally used to indicate catastrophic conditions in Z-80 systems. What could be more final in the TRS-80 than a reset? The TRS-80 uses NMI when the reset button is pressed. The reset switch is tied indirectly to the NMI line that goes into the Z-80. Non-maskable means that this interrupt can never be disabled; reset is always active.

When the NMI (reset) is received, the Z-80 saves the address of the current instruction in the stack, and transfers control to location 66H.

That's it. You can look on the NMI interrupt as a type of CALL to location 66H. After interrupt processing, the last instruction executed must be an RETN, a return from non-maskable interrupt (unless the stack is reinitialized).

The remaining three types of interrupts are maskable. The Z-80 interrupts must be enabled by first executing the enable interrupts (EI) instruction. The three interrupt types are defined by three interrupt modes set by the IM 0, IM 1 or IM 2 instructions. The default mode is interrupt mode 0.

Interrupt mode 0 is not used in the TRS-80. It is a mode compatible with the aging father of the Z-80, the 8080. In this mode, an interrupt signal from outside sets the INT* signal line. The CPU then asks "who's there?" by an interrupt acknowledge—a combination of the IORQ and M1 signals. The external device controller then responds by jamming a RST instruction onto the data bus lines. Aha! The RST is not only used for short CALLS to page 0 of ROM! The RST was originally developed as an external eight-bit interrupt code to permit eight separate interrupts to locations 0, 8, 10H, 18H, 20H, 28H and 30H. Can you use this interrupt sequence? Sure. Build an external device controller, bring out the IORQ and M1 signals on the bus, and put in new ROM and you're set.

Interrupt mode 2 is also not used in the TRS-80. In this mode the I register is loaded with the most significant byte of the address of an interrupt vector table. If the vector table were at E000H, the I register would be loaded with E0H, for example. The external interrupt would supply the INT* signal as before, with the CPU responding with an interrupt acknowledge.

The external device would then supply the lower-order eight bits of the vector table address, the CPU would assemble it into a 16-bit address, and a vectored CALL would be made via the proper vector table address. This mode permits 128 separate interrupts, as each vector table entry is a two-byte address. Can this interrupt mode be used in the TRS-80? Here again, the interrupt-acknowledge signals IORQ and M1 are not available on the TRS-80 bus.

The last interrupt mode is interrupt mode 1. This mode *is* used in the TRS-80. An interrupt is put on the INT* line as before. When the INT* is received, the CPU essentially performs a CALL to location 38H, a dedicated location to handle this interrupt. An interrupt acknowledge is unnecessary, as no external response is necessary.

Two devices in the TRS-80 use the mode 1 interrupt. The first of these is the real-time clock. The real-time clock is simply a divided-down clock frequency generator that activates the INT* line every 25 milliseconds. The second source of the mode 1 interrupt is an interrupt from the disk drive controller chip, the 1771. The two signal sources go onto data lines 7 and 6 respectively, so that the TRSDOS code can decide which condition caused the interrupt. See Fig. 1.

What are we left with after this little discussion? Two interrupts—the NMI (location 66H) and the mode 1 interrupt (location 38H). If you look in your disassembled Level II listing, you'll see that there's not much we can do with the NMI. It immediately starts out testing for the disk to reboot. New ROM code, anyone?

The mode 1 interrupt is a different story. The first thing that happens at location 38H is a jump to 4012H! If we can substitute our own code at 4012H. . .

That Blinkin' Interrupt Handler

To get our feet wet in the muddy waters of the interrupt morass, let's describe a simple interrupt processor. The program shown in Program Listing 1 is about the simplest one I could come up with. It blinks an asterisk on and off in the upper right hand corner of the screen. Each asterisk on time corresponds to 256 real-time clock interrupts; each blank corresponds to another 256 real-time clock interrupts.

To use the program, follow this sequence:

1) Assemble the program (optional) and output the object file to disk as INT/OBJ.
2) Load DOS and perform a LOAD INT/OBJ. Or, as an alternative, key in the machine code using DEBUG at 7F00H.
3) Load BASIC and protect memory by answering 32511 for MEMORY SIZE?.
4) Disable the interrupts by CMD"T" in BASIC. This command performs a disable interrupt (DI) instruction, to mask out real-time clock interrupts.
5) Enter POKE 16403,0:POKE 16404,127 in BASIC. This replaces the JP 4518H instruction normally found at location 4012H to a JP 7F00H.
6) Enter a CMD"R". This restarts the interrupts. You should now see a blinking asterisk in the upper right hand corner that blinks at the rate of 6.4 seconds.

The procedure above substitutes the INT interrupt handler before the TRSDOS handler. When the clock is turned on (EI instruction executed), a real-time clock interrupt causes a CALL to location 38H. Location 38H holds a JP 4012H. By substituting our own jump instruction at 4012H, our own interrupt handler is entered.

The INT routine first saves the A register and flags. This is absolutely necessary because we will be using A and the flags in the routine. No other registers are used, so saving AF suffices.

The routine bumps the count in CNT by one. If the count is zero, the last character of line 0 is Exclusive ORed with 0AH. If a blank was present previously, it is changed from 20H to 2AH by the XOR. If an asterisk was present, the 2AH is changed to a 20H. A POP restores the A register and flags, and the normal RTC interrupt routine at 4518H is entered.

We've used a JP to 4518H here so that the normal interrupt routine is entered after we do our own processing. This is necessary because the DOS routine process-

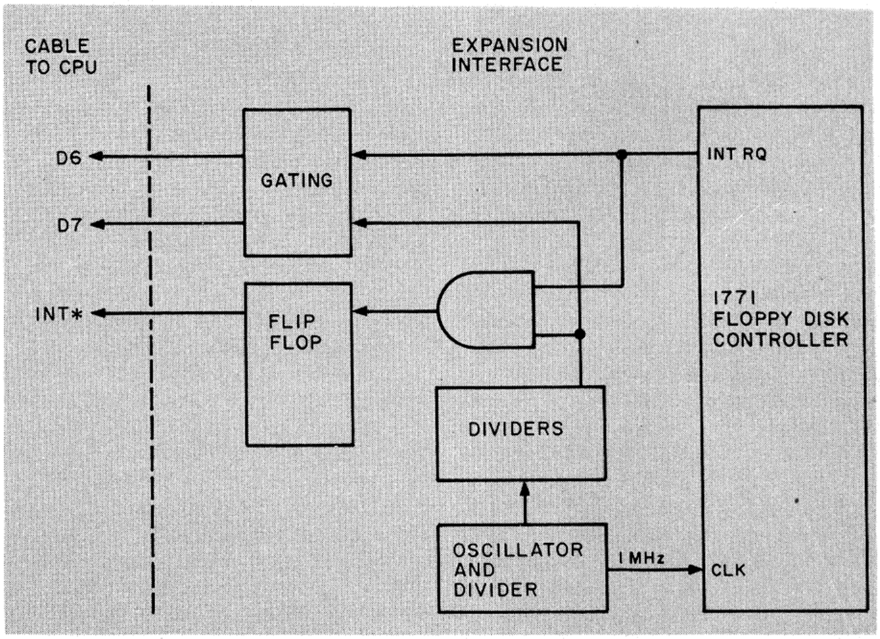


Fig. 1. INT Signal Generation*

```
F000          00100            ORG      0F000H          ;***CHANGE THIS***
              00110 ;*****************************************************
              00120 ;               INTERRUPT TRACE PROGRAM                  *
              00130 ; PRINTS OUT ONE TO SIX TRACE "WORDS" AT 25 MS INTERRUPT*
              00140 ;*****************************************************
              00150 ;
F000 F5       00160 INT     PUSH    AF              ;SAVE ENVIRONMENT
F001 D5       00170         PUSH    DE
F002 E5       00180         PUSH    HL
F003 DDE5     00190         PUSH    IX
F005 11003C   00200         LD      DE,3C00H        ;START OF LINE 0, SCREEN
F008 DD215DF0 00210         LD      IX,ADDTAB       ;START OF ADDRESS TABLE
F00C DD7E00   00220 INT010  LD      A,(IX)            ;GET LSB OF ADDRESS
F00F DDB601   00230         OR      (IX+1)            ;MERGE MSB
F012 281F     00240         JR      Z,INT020          ;GO IF DONE
F014 DDE5     00250         PUSH    IX                ;TRANSFER IX TO HL
F016 E1       00260         POP     HL
F017 23       00261         INC     HL                ;POINT TO MS BYTE
F018 CD3BF0   00270         CALL    MEMOUT            ;DISPLAY ADDRESS VALUE
F01B 3E3D     00280         LD      A,'='             ;EQUALS
F01D 12       00290         LD      (DE),A            ;DISPLAY =
F01E 13       00300         INC     DE                ;BUMP SCREEN POINTER
F01F DD6E00   00310         LD      L,(IX)            ;GET LSB OF ADDRESS
F022 DD6601   00320         LD      H,(IX+1)          ;MSB OF ADDRESS
F025 23       00321         INC     HL                ;POINT TO MS BYTE
F026 CD3BF0   00330         CALL    MEMOUT            ;DISPLAY CONTENTS
F029 3E20     00340         LD      A,' '             ;BLANK
F02B 12       00350         LD      (DE),A            ;DISPLAY BLANK
F02C 13       00360         INC     DE                ;BUMP POINTER
F02D DD23     00370         INC     IX                ;BUMP TABLE INDEX
F02F DD23     00380         INC     IX
F031 18D9     00390         JR      INT010            ;GO FOR NEXT ADDRESS
F033 DDE1     00400 INT020  POP     IX              ;RESTORE ENVIRONMENT
F035 E1       00410         POP     HL
F036 D1       00420         POP     DE
F037 F1       00430         POP     AF
F038 C31845   00440         JP      4518H           ;***THIS MAY CHANGE***
F03B CD42F0   00460 MEMOUT  CALL    ROTATE          ;OUTPUT 1ST AND 2ND DIGITS
F03E CD42F0   00470         CALL    ROTATE          ;OUTPUT 3RD AND 4TH DIGITS
F041 C9       00480         RET
F042 AF       00490 ROTATE  XOR     A               ;CLEAR FOR RLD
F043 CD4DF0   00500         CALL    ASCII           ;OUTPUT DIGIT
F046 CD4DF0   00510         CALL    ASCII           ;OUTPUT DIGIT
F049 ED6F     00520         RLD                     ;RESTORE VALUE
F04B 2B       00530         DEC     HL              ;BUMP POINTER TO LS BYTE
F04C C9       00540         RET
F04D ED6F     00570 ASCII   RLD                     ;GET FIRST 4 BITS
F04F F5       00575         PUSH    AF              ;SAVE UNALTERED DIGIT
F050 C630     00580         ADD     A,30H           ;CONVERT TO ASCII
F052 FE3A     00590         CP      3AH             ;CHECK FOR A-F
F054 FA59F0   00600         JP      M,AS010         ;GO IF 0-9
F057 C607     00610         ADD     A,7             ;CONVERT TO A-F
F059 12       00620 AS010   LD      (DE),A          ;STORE
F05A F1       00625         POP     AF              ;GET UNALTERED DIGIT
F05B 13       00630         INC     DE              ;BUMP PNTR
F05C C9       00640         RET                     ;RETURN
F05D D640     00650 ADDTAB  DEFW    40D6H           ;SET LAST ADDRESS TO 0
F05F E840     00660         DEFW    40E8H
F061 FD40     00670         DEFW    40FDH
F063 FFFF     00680         DEFW    0FFFFH
F065 FFFF     00690         DEFW    0FFFFH
F067 FFFF     00700         DEFW    0FFFFH
F069 0000     00710         DEFW    0               ;DON'T CHANGE THIS!
0000          00720         END
00000 TOTAL ERRORS
```

Program Listing 1. Interrupt Blinker Program

es interrupts caused by the disk when the clock is on.

If you want to bypass the DOS routine completely, you'll have to turn off the clock for disk operations and substitute the following code:

```
OUT  LD    A,(37E0H)  ;RESET INTERRUPT
     POP   AF         ;RESTORE REGS
     EI               ;ENABLE INTERRUPTS
     RET1             ;RETURN FROM INTERRUPT
```

The LD A,(37E0H) instruction resets the expansion interface logic that caused the interrupt. If this weren't done, another immediate interrupt would result after the return from interrupt. POP AF restores the registers as before. EI re-enables the interrupt. The interrupts were automatically disabled by the CPU when the interrupt sequence was started and remain off until a new EI is performed. RETI returns from the interrupt in an analogous operation to a RET from subroutine; it pops the return address from the stack and loads it into the PC register.

The asterisk will keep blinking in similar fashion to the real-time clock display, through BASIC program loads and execution, and returns to DOS and DOS commands.

The Case of the Strange Reset

You'll notice in the above code that an LD A,(37E0H) was performed to reset the flip-flop for the pending real-time clock interrupt. If this weren't done, another interrupt would occur directly after the EI and RETI, and another, and another.

In the process of checking out the Blinker, I found that the interrupts seemed to occur at double the rate expected. I was getting 5120 interrupts in 64 seconds, making the rate 80 interrupts per second rather than 40. This rate occured in both the stand-alone version of Blinker and the version that jumps to the normal DOS interrupt handler.

An oscilloscope verified that the rate was indeed 25 milliseconds. Where was the problem?

I looked at the expansion interface schematic again. Yes, a 40 Hertz signal came out of the divider chain and went into two strangely connected flip-flops. At this point I called the local TRS-80 hardware guru, Dan Likins.

Dan quickly found the answer. "It looks like you need *two* reads of 37E0H to reset the logic," he exclaimed. Sure enough, with two LD A,(37E0H)s, Blinker runs at the expected rate of 40 interrupts per second.

```
7F00          00100            ORG      7F00H           ;CHANGE THIS FOR YOUR SYSTEM
              00110 ;*****************************************************
              00120 ;*               INTERRUPT BLINKER                    *
              00130 ;*****************************************************
              00140 ;
7F00 F5       00150 INT     PUSH    AF              ;SAVE ENVIRONMENT
7F01 3A167F   00160         LD      A,(CNT)         ;GET COUNT OF TIMES
7F04 3C       00170         INC     A               ;BUMP
7F05 32167F   00180         LD      (CNT),A         ;STORE
7F08 2008     00190         JR      NZ,OUT          ;GO IF NOT 256TH TIME
7F0A 3A3F3C   00200         LD      A,(3C3FH)       ;GET LAST CHAR OF LINE 0
7F0D EE0A     00210         XOR     0AH             ;CHANGE SPACE/ASTERISK
7F0F 323F3C   00220         LD      (3C3FH),A       ;STORE IN LINE
7F12 F1       00230 OUT     POP     AF              ;RESTORE REGS
7F13 C31845   00240         JP      4518H           ;GO TO RTC INT HANDLER
7F16 00       00250 CNT     DEFB    0               ;INITIALIZE TO 0
0000          00260         END
00000 TOTAL ERRORS
```

Program Listing 2. Interrupt Trace Program

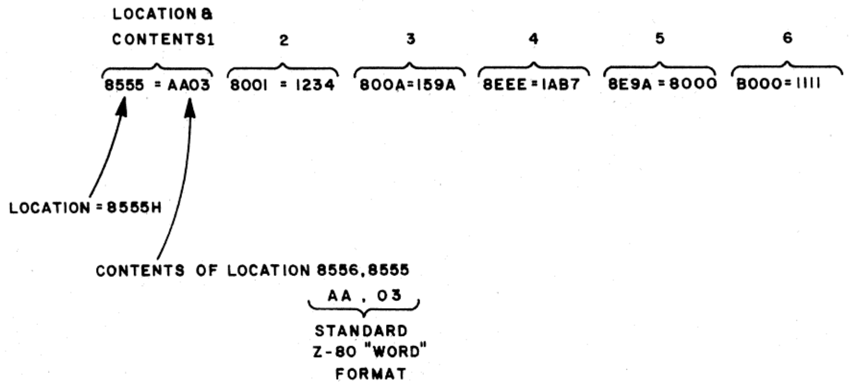


Fig. 2. Trace Display

Dan and I mulled over the design philosophy. One thing appears obvious to me—the DOSes as implemented now process two real-time clock interrupts, one immediately following the other, until the logic is reset by the second LD A,(37E0H). I would be interested in comments from any reader who has worked on this problem.

More Ambitious Interrupt Handlers

I stated before that interrupt handlers could be hundreds of instructions long. That's not always true. Interrupt handlers are usually written in tight code to be as fast as possible. The constraint here in the clock interrupt is that it must complete before the next 25 ms interrupt, which amounts to about 25,000/5 = 5000 instructions. Of course, 5000 instructions wouldn't leave much time for any other processing.

We can use the real-time clock interrupt, however, to implement any short foreground function. Suppose that we had the TRS-80 hooked up to our home "computer central" that was monitoring burglar alarm data, room environment, and weather data. Every 25 milliseconds we could poll the inputs (with some simple additional hardware) to check on the status of our inputs and take appropriate actions. At the same time we could be entering and executing BASIC programs! The foreground interrupt processing would be automatic and transparent.

Well, almost transparent. BASIC and TRSDOS use software for timing such things as keyboard debounce and cassette bit times. With a large interrupt processing program, some characters may be missed on keyboard entry. You'll have to turn off the clock on loading cassette tapes, and you'll be disabling your foreground processing during that time. However, the potential is there for doing short foreground processing.

An Interrupt Trace Program

As an example of a more ambitious foreground task, consider Program Listing 2. It is an interrupt trace program. It continuously displays one to six selected locations on the first line of the screen. The six locations are assumed to be words in standard Z-80 word format, as shown in Fig. 2.

The six locations to be displayed are entered into the ADDTAB table. The first zero entry marks the end of the table. In other words, to display one location, put the address of the location in the first two bytes of ADDTAB and zeros in the next two.

The display appears continuously on the first line of the screen as long as the clock is enabled. Follow the identical loading instructions as in the previous program, except of course, loading into F000H, protecting 61439, and POKEing 240 instead of 127. The table addresses can be changed at any time by POKEing addresses via BASIC.

One of the more interesting things to trace is the string work area pointer at 40D6H. The string handling ability of BASIC is well known, and is characterized by those interminable pauses while BASIC heals itself by compacting and cleaning up strings. You can actually see this string cleanup by displaying 40D6H in the

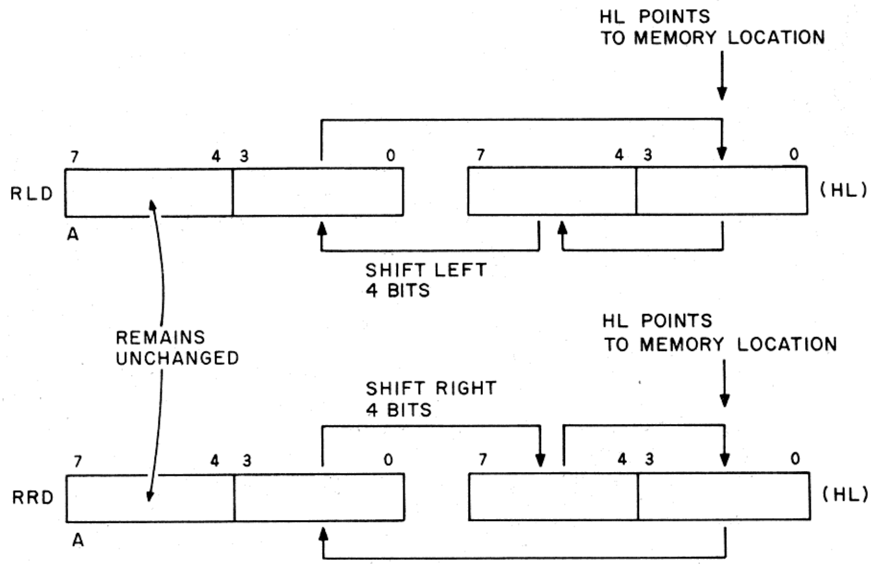


Fig. 3. RLD and RRD Instructions

Location	Locations Displayed	Contents	Display Was	Display Should Have Been
1	2,1	AF	0001 = CCAA	0001 = C3AF
2	3,2	C3	0002 = 77CC	0002 = 74C3
3	4,3	74	0003 = 0077	0003 = 0674
4	5,4	06	0004 = CC00	0004 = C306
5	6,5	C3	0005 = 00CC	0005 = 00C3
6	7,6	00	0006 = 4400	0006 = 4000
7		40	—	—
		Answer = ?		

Table 1. Trace Debug Problem

✓ Reader Service—see page 242

trace. Other suggested pointers are the stack pointer at 40E8H and the start of free memory pointer at 40FDH.

The program itself is not too complicated, with one exception. All registers used in the routine are saved in the stack. DE is initialized with the start of the screen memory and IX with the ADDTAB start.

The code with the indented comments is the main driver loop. It first checks for a zero entry in the ADDTAB table. When found, the last location has been displayed, and it exits to normal interrupt processing at INT020.

If a valid address (non-zero) is found in ADDTAB, the address value is displayed by calling MEMOUT which calls ROTATE, which inturn calls ASCII. An equals sign is then displayed and the contents of the address value are then converted in a similar manner. After this conversion, a blank is displayed, and the driver loops back to INT010 for the next address from ADDTAB.

This routine makes use of the RLD instruction, which is a unique BCD shift. The shift operates four bits at a time between the least significant four bits of A and the contents of the memory location pointed to be the HL register pair, as shown in Fig. 3. An RRD operates similarly for a right shift.

During the process of debugging, I found an unusual problem. The ADDTAB value was being converted properly, but the contents of the address were not being converted correctly. Specifying 1234H as the first address of ADDTAB resulted in 1234= on the screen, but this was followed by 3344 instead of the actual contents of 3547. To debug, I made a list of the locations to be displayed, the resulting display, and the correct display (shown in Table 1). Can you see what I did wrong in the program or in my mind? The first five correct replies will receive a copy of my Howard W. Sams book *Z-80 Microcomputer Design Projects*.

Disk Dump Subroutine

Here's a clarification on my November column. The code at FE72H should consist of 41H, 4CH, 49H, 4EH, 45H, followed by 18 blanks (20H), 0DH, followed by eight blanks (20H). This pads out the file name properly. It is correct as it stands, but not very obvious since only the first character of a DEFM is shown.

Next month we'll have more assembly language topics. If you have ideas on what you'd like to see, write me at 28182 Palmada, Mission Viejo, CA 92692. ■

INSIDE 80

Continued from p. 8

We tell our field people only what we're very sure of. And still, we blow it sometimes. Speculation leads to customer ill-will every time.

Advance information on new products is not made available for a number of reasons. For instance, we might have to make a price or specification change at the last minute. Then there's the threat of unforseen delay of weeks or even months. What if a product must be cancelled? Where, I ask you, is the poor customer who invested time and money in a TRS-80, on the strength of a canceled printer we had promised in 90 days? I hope you can see the wisdom of knowing the facts before we open our corporate mouths—even to our own people.

Model II Owners—Help!

In most Model II software packages, there is an Owner Registration Card (look in the back of your binder). Please fill in the card, and return it to us. Be sure the catalog number of the software package is correctly entered in the upper right-hand corner. We get them back with numbers of anything—even printer cable catalog numbers.

We're building a file of owners so we can let you know when a bug is found, or an enhancement is made to the specific package(s) you own. Without your correct address, we can't let you know. So far it appears that the card return rate is low. If you haven't sent yours, do so.

Pocket Computers

Back in the November/December months, Pocket Computers were a bit hard to find. Plentiful quantities should now be reaching the warehouses. So, if you had trouble finding a TRS-80 Pocket Computer, try us again. ■

80 APPLICATIONS

by Dennis Kitsz

"A section of memory slightly more than 2,000 bytes long was left unassigned by Tandy engineers, presumably for future expansion."

The TRS-80 is already something of a legend, and will probably be remembered as the Univac of personal computers—not the first, but simply the best-known representative of its age. No, this is not a eulogy for the Model I, but instead, a bit of praise for a machine all too often disparaged as the "Trash-80." Through its versatility, it has proven the viability of personal computers.

Many of you are familiar with the memory map of the Model I, but here's a brief review. Of the 65536 possible memory locations in the 80, 12,288 are reserved for the BASIC language in Read Only Memory (ROM). A handful of addresses serve as windows to the dual-cassette latch, printer, and disk drives. The first block of 16,384 bytes of read/write memory is broken up into two parts—a section reserved for use by the computer's operating system, and the rest available to the user for BASIC programming. Two additional 16K blocks of programmable memory can be added in an expansion unit.

A section of memory slightly more than 2,000 bytes long was left unassigned by the Tandy engineers, presumably for future expansion. The new Model III uses this unassigned space for extensions and changes to the BASIC language of the Model I. It can be assumed that Radio Shack will not provide future expansions in that area. This month's Applications will show how to use that memory space for data storage, utility programs, and as a development system. The Memory Sidecar implements three different concepts on a single circuit board: 2K of user RAM, 2K of user ROM, and a special approach to developing future ROM-based programs I call "read-only-RAM".

Read-only-RAM is true to its name. Once a program has been written into this block of memory, a switch can be flipped which will not permit the information to be changed. In this way, programs which will eventually be burned into ROM can be fully emulated by this non-writable RAM—a kind of digital try-before-you-buy.

For this project you will need a power supply and ten integrated circuits: two 74LS00, one 74LS02, one 74LS30, one 74LS125, four 2114 memory chips, and one 2716 2K EPROM (5-volt only). Of the last circuits, because of their cost ($5 each for the 2114's, and about $20 for the 2716), you may wish to install only 1K of RAM to start with—two 2114's. You will need all the 74LS circuits for this project, and I would recommend sockets for all parts. A special socket known as a zero-insertion-force socket is used for the ROM; it has a handle which squeezes its contacts apart so ROMs can be inserted or removed without bent pins. (See the parts list for prices of a complete set of parts, or assembled and tested versions.)

The first task of the Memory Sidecar is to decode the addresses which will identify the locations of its memory. Those unassigned addresses are 3000 to 37DF (12288 to 14303 decimal). Z2a/b and Z3a/b partially decode the area from 3000 to 37FF, and Z1 together with Z2c decode an address portion defined as 07E0. These two signals, (the output of Z3b and Z2c) when combined by Z2d, allow all address signals from 3000 to 37DF to flow through to the output of Z2d. (Exatron Stringy-Floppy owners note: since Exatron has gobbled up this memory space without allowing you to patch the ESF out of the operating system, the Memory Sidecar cannot be used without powering down or disconnecting the ESF unit.)

This decoding signal would be sufficient to select a 2K EPROM like the 2716. Unfortunately, similar 2K RAMs are expensive and hard to get, so I chose the type 2114 1K RAMs instead. Address line 10 is buffered (by Z5d) and combined with the select line in Z4c to produce 3400 to 37DF; it is inverted (by Z4a) and combined with Z4b to produce 3000 to 33FF. This way, each bank of 1K RAM has its own select signal.

The outputs of S4b, c, and d, respectively, select low RAM, high RAM, and all ROM. Two sets of signals will compete for the attention of the CPU if both ROM and a bank of RAM are turned on at the same time. To avoid this, their respective select signals are fed through the three-state buffers in Z5. The individual buffers in Z5 are turned on according to the position of S2.

The signal from S2 can select RAM or ROM only if the proper address occurs simultaneously with a READ or WRITE signal from the computer. Assuming switch S1 is closed, either a low READ or low WRITE signal will cause Z3c to go high, and it is in turn inverted to a low signal by Z3d. This condition will activate the appropriate Z5 buffers. If the output of one of the Z5 buffers is low, the appropriate block of memory will be selected.

S1 is the heart of the read-only-RAM idea. When it is closed, all signals proceed normally as described above. But when S1 is open, the 1K resistor locks the input of Z3c high. Thus, although the RAM receives the memory-write command from the computer, the command is disconnected from its address-select circuit. In other words, the RAM will not believe it is being asked to store data!

Fig. 2 presents the wiring of the memory circuits themselves. The select signals are fed to each of the ICs, as are the address lines. Nine address lines select data from the 2114's, and ten lines are needed for the 2716. Each 2114 is a 1K by 4-bit circuit, so two are used for a full 1K by 8 block of bytes. Note that lines D0 through D3 are fed to Z6 and Z8, and lines D4 through D7 are connected to Z7 and Z9. The 2716 is a 2K by 8 bit device, so all eight data lines are attached to it.

Fig. 3 is the optional (but recommended) data bus termination. Because 26 signal-carrying lines are used by the Memory Sidecar, a great amount of local electronics noise is produced. This noise has the potential to cause program crashes (particularly if you are wire-wrapping this circuit), so I recommend terminating each data line with a 1k-ohm resistor to ground. In fact, this improved the reliability of my system as a whole, stabilizing the data transfer mechanisms throughout the TRS-80.

The entire device can be fit on a three-inch by five-inch board, as shown in Photo 1. Because of the number of connections, I recommend wire-wrapping as opposed to soldering (Photo 2).

Giving It a Test

With your system's power off, connect

NEVER UNDERSOLD!

That's right, if you can find a lower price in this magazine for any of the items listed in this ad, we will reduce our price below our competitor's price. See each box below to determine how much EXTRA we will cut off of THEIR price if we're not lowest. Please consider the competitor's shipping charges, OUR SHIPPING IS FREE!*

FLOPPY DISKETTES & SUPPLIES

IF YOU CAN FIND A LOWER PRICE IN THIS MAGAZINE ON ANY OF THESE ITEMS, DEDUCT $.50 FROM OUR COMPETITOR'S PRICE, THAT'S OUR PRICE!

Call For Quantity Discounts

Verbatim Diskettes (box of 10)

5¼" MD525-01 soft, 10 or 16	$26.50
5¼" MD577-01 quad soft, 10 or 16	$33.00
8" FD34-1000 soft	$30.00
8" FD32-1000 hard	$30.00
8" FD34-8000 double density soft	$44.00
8" FD32-8000 double density hard	$44.00

Printwheels (specify style)

Qume or Diablo	$6.50

Labels

3½"x15/16" (5000 labels)	$18.75
Other sizes and quantities	CALL

Ribbons

Diablo Hy Type I	$4.95
Diablo Hy Type II	$5.25
Qume Sprint	$3.50
Centronics Zip Pack	$3.95
MANY OTHERS	CALL

PRINTERS

IF YOU CAN FIND A LOWER PRICE IN THIS MAGAZINE ON ANY OF THESE PRINTERS, DEDUCT $10 FROM OUR COMPETITOR'S PRICE, THAT'S OUR PRICE!

Paper Tiger IDS-440G	$939
Paper Tiger IDS-460G	$1193
Anadex DP-8000	$855
Anadex DP-9500	$1395
TI-810 Basic	$1625
Centronics 737	$825
NEC 5500 D w/ Bidrctnl Board	$2695
NEC 5530	$2595
VISTA Daisey Wheel Printer	$1834
Qume 5/45,5/55	CALL
Escon IBM Interface	$595

Call For Other Printers

HARDWARE

IF YOU CAN FIND A LOWER PRICE IN THIS MAGAZINE ON ANY OF THESE ITEMS, DEDUCT 5% FROM OUR COMPETITOR'S PRICE, THAT'S OUR PRICE!

Novation CAT Modem	$145
Novation D-CAT Modem	$185
16K Memory kit	$46
Isolators	$49
Shugart 35tr Drive	$349
Pertec or MPI 40tr Drive	$359
Lobo Drives	CALL
Matchless Drives	CALL
Percom Doubler	$209
Percom Seperator	$27
AIM-65 Computer	375
TI-99/4 Computer	$925
California Computer Systems Bds	CALL
Symtec Computer Boards	CALL
Mountain Hardware Boards	CALL
Green Screen	$11

Call for Other Hardware

4636 Park Granada
Calabasas, Ca. 91302

✓69

Alpha Byte Storage

VISA

For phone orders CALL:

(213) 883-8594

SPECIAL #1

If you purchase the "TRS-80 DISK AND OTHER MYSTERIES" Book for the regular price of $22.50
you can buy 10 VERBATIM DISKETTES AND a plastic library case for $22.00
TOTAL $44.50

SPECIAL #2

If you purchase APPARAT NEWDOS+ for the regular price of $99.95
you can buy 10 VERBATIM DISKETTES AND a plastic library case for $5.00
TOTAL $104.95

SPECIAL #3

If you purchase APPARAT NEWDOS/80 for the regular price of $149.00
we will give you 10 VERBATIM DISKETTES AND a plastic library case .... FREE
TOTAL $149.00

SPECIAL #4

If you purchase the MICROSOFT BASIC COMPILER for the REDUCED PRICE of $190.00
we will give you 10 VERBATIM DISKETTES AND a plastic library case .... FREE
TOTAL $190.00

SPECIAL #5

If you purchase the new "MICROSOFT BASIC REVEALED" book for the regular price of $29.95
you can buy 10 VERBATIM DISKETTES AND a plastic library case for $20.00
TOTAL $49.95

SPECIAL #6

SOFTWARE DEALS--If you want to purchase any software (including Utilities, Operating Systems, Games, Buisness Programs, etc.) we will automatically deduct from 10% to 40% off the regular retail price.

*FREE shipping on all orders over $20. Visa and Master Card accepted. All never undersold offers good as supply lasts. Please add 2.00 for all COD orders. Please call for items not listed. We gladly answer any questions on all of our hardware, software, and supply needs. Quanitity discounts available. School purchase orders accepted. Please remember to figure competitors shipping and handling charges when arriving at never undersold price.

the Memory Sidecar to your TRS-80. Set S1 to RD/WR, and S2 to RAM. Power up the circuit, then your TRS-80, and follow through the power-up sequence. As with all homebrew attachments to your computer, *if it does not act normally turn the 80 off immediately!* Then recheck your work.

If all seems right, enter the following short program:

```
10 Y = 15360
 20 FOR X = 12288 TO 13311
 30 POKE X, PEEK (Y)
 40 Y = Y + 1
 50 NEXT
 60 Y = 15360
 70 CLS
 80 FOR X = 12288 TO 13311
 90 POKE Y, PEEK (X)
100 Y = Y + 1
110 NEXT
```

Whatever currently appears on you screen should be transferred to the low block of the Memory Sidecar. The screen will clear, and the transferred information should then be put back on the screen.

If it works, shout and stomp (I did that), and replace lines 10 and 60 with FOR X = 13312 TO 14303. Run it again to test the high block of memory. If either test doesn't work, check these symptoms:

- Program crash and/or return or MEMORY SIZE?

  One or more of your data or address lines are incorrectly wired. You may have left out the ground connection (computer pin 37). The RD or WR lines may be incorrectly wired.
- Program locks up but does not crash; can be reset.

  The address lines may be wired in incorrect order, and you are inadvertently selecting two addresses at once.
- Screen fills with graphics blocks after the CLS.

  No memory is being read. Check the positions of the switches and the power connections. Also, double-check the wiring of the RD and WR lines, and Z5. If you are only using two of the 2114's, insert them in the other sockets (with the

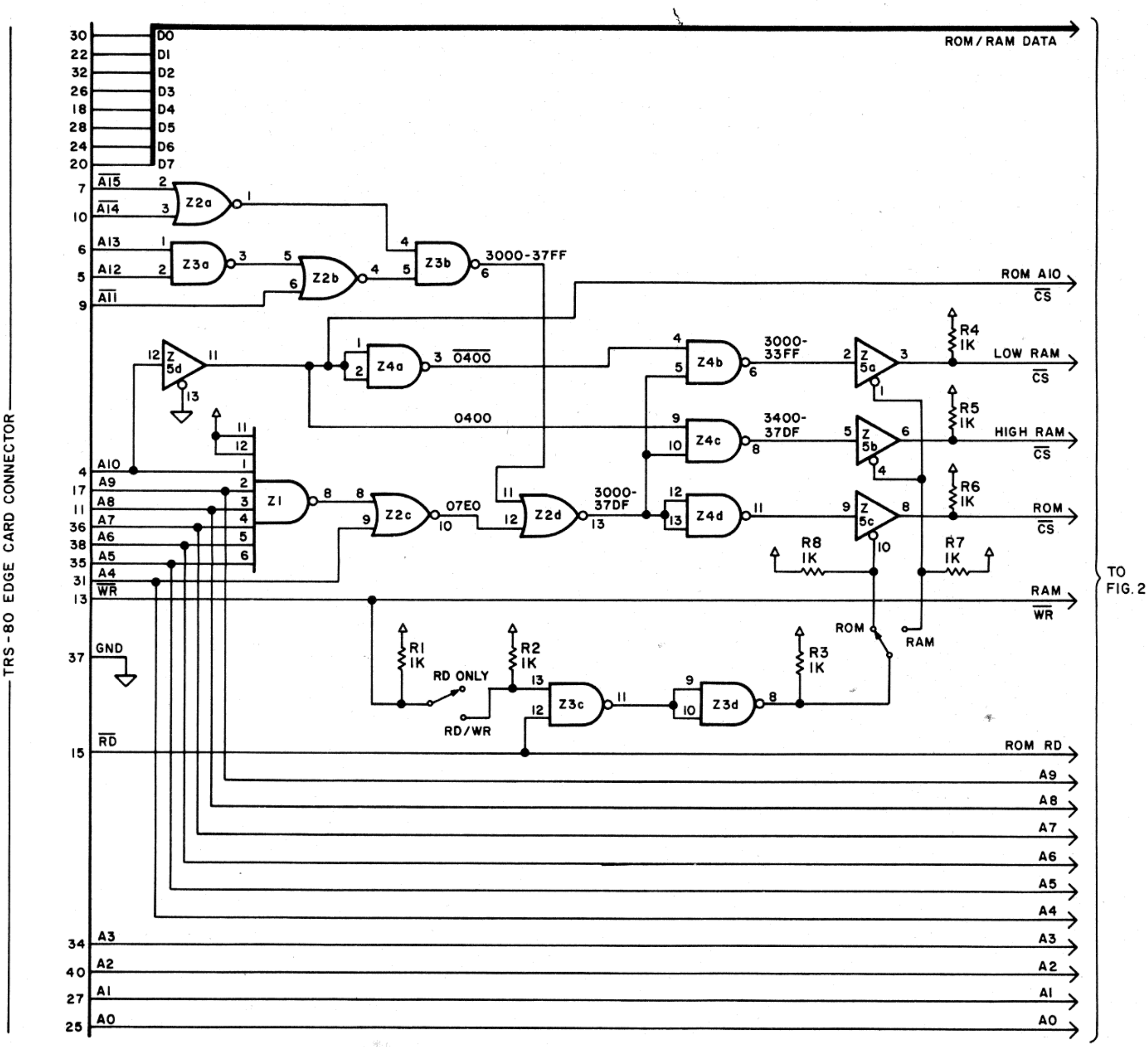


Figure 1

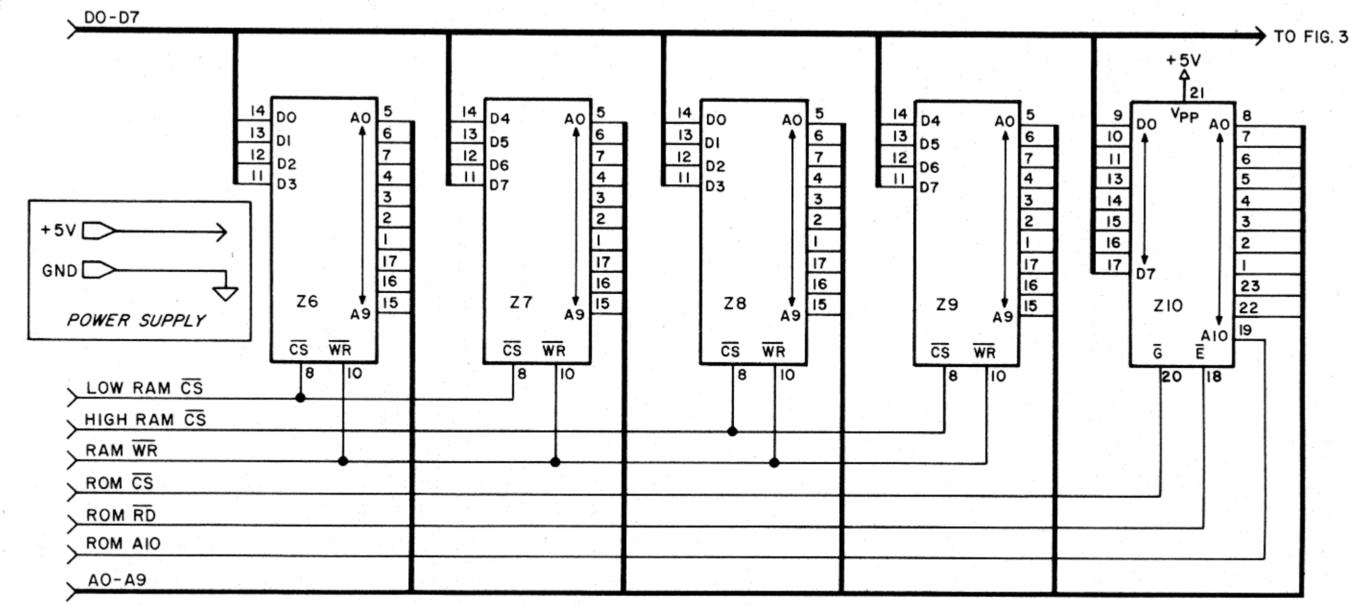


Figure 2

power off!) and try again.

- Screen shows incorrect characters, correct spacing.

  Data lines are incorrectly or partially wired. If you are using only two 2114's, one may be in the wrong socket.

- Screen fills with garbage.

  This is a tough one. You may have improperly wired the address lines or the chip selects, meaning you have written garbage to the memory, or you have read from some other area of memory. Double check the address wiring.

> "The TRS-80 is already something of a legend . . . not the first, but simply the best-known representative of its age."

Once you get a proper result from the board, flip S1 to read only position, change whatever you have on the screen, and run the program again. What comes back to the screen should be the old screen you wrote before you changed it. In other words, the program is reading the memory, but the POKEing has had no effect. This is the object of read-only-RAM.

Finally, if you have any programmed 2716 EPROMs, insert one carefully into its socket. Flip the switch to the ROM position, and GOTO 45 in the program above. The contents of the 1K of ROM should be written to the screen. Again, if you don't get what you expect, look for a wiring error in the ROM's address and data lines.

IC	*TYPE*	*VCC*	*GND*
Z1	74LS30	14	7
Z2	74LS02	14	7
Z3	74LS00	14	7
Z4	74LS00	14	7
Z5	74LS125	14	7
Z6-9	2114	18	9
Z10	2716	24	12

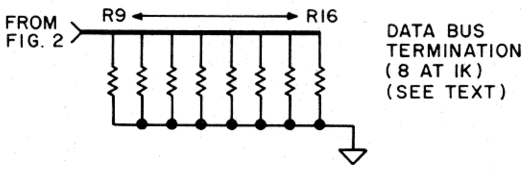


Figure 3

What To Do With It

Now the big test—a short program for encrypting data according to a simple random offset scheme. Listing 1 presents a BASIC program to create a new set of strings from the original set of strings which represent data. These new strings are made up of characters substituted for the originals. For example, THIS IS A TEST might become XuA/ L? m 0#*; when characters are substituted. Such a code is not easily broken, because the code is made up of whatever random numbers are in your private ROM. For simple encoding of game information or moderately confidential information, it is ideal. A code tape can be read into RAM, or better yet, a packet of code EPROMs can be kept under lock and key.

To start, POKE random numbers (0 to 255) into locations 12288 to about 12400. This will provide a random distribution of replacement offsets. PEEK back into those locations, and record the values you

Listing 1. BASIC encryption program using data at 3000 hex.

```
4016              00100          ORG     4016H
4016  0030        00110          DEFW    ENTREE
3000              00120          ORG     3000H
06CC              00130 BASIC2   EQU     06CCH
4036              00140 KEYHLD   EQU     4036H
3801              00150 KEYBRD   EQU     3801H
401A              00160 HOLDER   EQU     401AH
4099              00170 INKEYS   EQU     4099H
0060              00180 DELAYS   EQU     0060H
```

```
3ØØØ E5              ENTREE  PUSH   HL
3ØØ1 21ØB3Ø                  LD     HL,START
3ØØ4 22164Ø                  LD     (4Ø16H),HL
3ØØ7 E1                      POP    HL
3ØØ8 C3CCØ6                  JP     BASIC2
3ØØB 21364Ø          START   LD     HL,KEYHLD
3ØØE Ø1Ø138                  LD     BC,KEYBRD
3Ø11 16ØØ                    LD     D,Ø
3Ø13 ØA              CHKKEY  LD     A,(BC)
3Ø14 5F                      LD     E,A
3Ø15 A3                      AND    E
3Ø16 2Ø18                    JR     NZ,CKPREV
3Ø18 77                      LD     (HL),A
3Ø19 14              INCD    INC    D
3Ø1A 2C                      INC    L
3Ø1B CBØ1                    RLC    C
3Ø1D 79                      LD     A,C
3Ø1E D68Ø                    SUB    8ØH
3Ø2Ø 2ØF1                    JR     NZ,CHKKEY
3Ø22 Ø6Ø7                    LD     B,7
3Ø24 2D              DECL    DEC    L
3Ø25 86                      ADD    A,(HL)
3Ø26 1ØFC                    DJNZ   DECL
3Ø28 A7                      AND    A
3Ø29 3EØØ                    LD     A,Ø
3Ø2B CØ                      RET    NZ
3Ø2C 321A4Ø                  LD     (HOLDER),A
3Ø2F C9                      RET
3Ø3Ø A6              CKPREV  AND    (HL)
3Ø31 281F                    JR     Z,STORE
3Ø33 3A994Ø                  LD     A,(INKEYS)
3Ø36 A7                      AND    A
3Ø37 2ØEØ                    JR     NZ,INCD
3Ø39 3A1A4Ø                  LD     A,(HOLDER)
3Ø3C 3C                      INC    A
3Ø3D 321A4Ø                  LD     (HOLDER),A
3Ø4Ø FEFF                    CP     ØFFH
3Ø42 28Ø9                    JR     Z,DECA
3Ø44 C5                      PUSH   BC
3Ø45 Ø6FF                    LD     B,ØFFH
3Ø47 ØØ              TMWSTE  NOP
3Ø48 1ØFD                    DJNZ   TMWSTE
3Ø4A C1                      POP    BC
3Ø4B 18CC                    JR     INCD
3Ø4D 3D              DECA    DEC    A
3Ø4E 321A4Ø                  LD     (HOLDER),A
3Ø51 7B                      LD     A,E
3Ø52 73              STORE   LD     (HL),E
3Ø53 C5                      PUSH   BC
3Ø54 Ø1ØØØ2                  LD     BC,2ØØH
3Ø57 CD6ØØØ                  CALL   DELAYS
3Ø5A C1                      POP    BC
3Ø5B ØA                      LD     A,(BC)
3Ø5C A3                      AND    E
3Ø5D C8                      RET    Z
3Ø5E C5                      PUSH   BC
3Ø5F E5                      PUSH   HL
3Ø6Ø F5                      PUSH   AF
3Ø61 Ø64Ø                    LD     B,4ØH
3Ø63 3A3D4Ø                  LD     A,(4Ø3DH)
3Ø66 E6FD                    AND    ØFDH
3Ø68 67                      LD     H,A
3Ø69 F6Ø2                    OR     2
3Ø6B 6F                      LD     L,A
3Ø6C 7D              LOOP    LD     A,L
3Ø6D D3FF                    OUT    (ØFFH),A
3Ø6F 7C                      LD     A,H
3Ø7Ø D3FF                    OUT    (ØFFH),A
3Ø72 C5                      PUSH   BC
3Ø73 Ø64Ø                    LD     B,4ØH
3Ø75 1ØFE                    DJNZ   $+Ø
3Ø77 C1                      POP    BC
3Ø78 1ØF2                    DJNZ   LOOP
3Ø7A F1                      POP    AF
3Ø7B E1                      POP    HL
3Ø7C C1                      POP    BC
3Ø7D C3FBØ3                  JP     Ø3FBH
Ø6CC                         END    BASIC2
ØØØØØ TOTAL ERRORS
```

find. This is the code for this particular session. If you are using a ROM, you would not need to copy them down because they would be permanently "burned" into ROM. Once you have the numbers POKEd and recorded, use Listing 1 to encode some sample text. The data will take some time to record if the message is long.

If the coded tape is read back through normal INPUT#-1 statements, or if loaded onto the screen with a 500-baud tape copier, the visible data will be garbage. But, when read via the creating program, the offset will be removed and the data will be restored to its original condition at entry.

Listing 1 is very elementary, and may become useful only when incorporated in a more sophisticated program. The heart of the encoding is the subroutine at line 1000: The individual characters in the array D$ are created by first obtaining the ASCII value of the keyboarded character A$. An offset is created next. By taking the current position of the D$ array, and equivalent position from the beginning of the ROM (at 12288) can be identified. A value is PEEKed from this location, ANDed with 127 (to suppress a possible overflow greater than 255), and finally added to the ASCII value of the keyboarded character A$. This is the final coded information.

A more likely use for the Memory Sidecar is for storing your frequently used routines, such as KBFIX, and RS-232 driver, monitor program, lowercase driver, etc. If you have loaded a half dozen taped utilities before every session in the past, you'll probably appreciate the convenience of typing something like

```
SYSTEM
*?/12288
```

to activate all your routines at one time. Listing 2 presents my own KBEEPFIX program originally published in *80 Microcomputing* a year ago. This version has modifications made by Jack Decker, and has been relocated to the beginning of the Memory Sidecar's addresses. You can test its convenience by loading it into RAM using T-Bug. After loading, flip the switch to Read Only, and the data will be protected no matter how many times you return to the MEMORY SIZE? routine. To activate KBFIX, simply type SYSTEM /12288 as noted above.

The potential uses of the Memory Sidecar are as extensive as your normal RAM and your BASIC language, with the plus that its contents are always protected from normal actions by BASIC. I trust that this project will satisfy your needs, especially all of you who took the time to write with your suggestions.■

```
10 CLS:CLEAR500:DIMD$(100):PRINT"TYPE 1 TO CODE, 2 TO DECODE"
20 A$=INKEY$:IF A$="1" THEN 30 ELSE IFA$="2" THEN 200 ELSE 20
30 X=0:PRINT"ENTER INFORMATION TO BE ENCRYPTED":PRINT:PRINT
40 A$=INKEY$ : IF A$="" THEN 40
50 PRINTA$; : IF ASC(A$)>31 AND ASC(A$)<127 THEN GOSUB 1000
60 IF X=80 THEN 80
70 IF A$=CHR$(13) THEN 80 ELSE 40
80 D$(X)="/":D$(X+1)="/":D$(X+2)="/":D$(X+3)="/":D$(X+4)="/"
100 PRINT:PRINT"DATA HAS BEEN ENCODED; READY TAPE"
110 PRINT"PRESS ENTER WHEN READY"
120 A$=INKEY$ : IF A$="" THEN 120
130 FOR N=0 TO X STEP3 : PRINT#-1,D$(N),D$(N+1),D$(N+2)
140 NEXT : PRINT#-1,"<STOP>","<STOP>","<STOP>"
150 PRINT"DATA HAS BEEN SENT TO TAPE"
160 PRINT"PRESS 1 TO CONTINUE, 2 TO END"
170 A$=INKEY$ : IF A$="1" THEN RUN
180 IF A$="2" THEN CLS : END
190 GOTO 170
200 CLS : ON ERROR GOTO 250
210 PRINT"LOAD TAPE INTO RECORDER; ENTER WHEN READY"
220 A$=INKEY$ : IF A$="" THEN 220
230 INPUT#-1,D$(N),D$(N+1),D$(N+2)
240 IF D$(N)="<STOP>" THEN 250 ELSE N=N+3 : GOTO 230
250 ON ERROR GOTO 290 : PRINT"DATA LOADED - NOW DECODING:"
260 PRINT:FOR X=0 TO N : IF D$(X)="" OR D$(X)="<STOP>" THEN 290
270 PRINTD$(X); : D$(X)=CHR$(ASC(D$(X))-(PEEK(12288+X)AND127))
280 NEXT X : PRINT : PRINT : ON ERROR GOTO 290
290 PRINT:PRINT:PRINT"DECODED INFORMATION FOLLOWS:":PRINT
300 FOR X=0 TO N : PRINT D$(X); : NEXT
310 PRINT:PRINT"PRESS 1 TO CONTINUE, 2 TO END"
320 GOTO 170
1000 D$(X)=CHR$((PEEK(12288+X)AND127)+ASC(A$)):X=X+1:RETURN
```

Listing 2. KBEEPFIX for the Memory Sidecar, with modifications by Jack Decker.

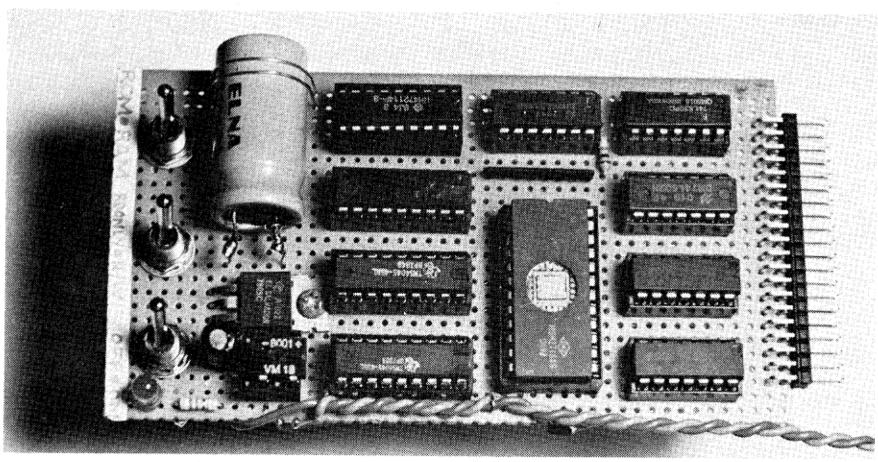
IMPORTANT BOX!!!!!

There are four ways to obtain memory addition devices for you Model I. The first is to build the project as shown in this column. The second is to obtain a complete, assembled and tested version available from The Peripheral People, P.O. Box 524, Mercer Island, Washington 98040. This model is being manufactured from the designs presented in this column, and costs $149 complete with a zero-insertion-force socket to hold a 2716 EPROM (not included), 2K or RAM, and power supply. The main circuit board alone can be obtained for $24.95.

Programmed ROMs can be purchased from Personal Microcomputers, Inc., 475 Ellis Street, Mountain View, California 94043. Many readers may recognize Personal Microcomputers as the manufacturer of the PMC-80, which is described as a "work-alike" version of the TRS-80, containing an identical Level II BASIC and software compatibility. The first ROM available from PMC to fit the circuit in this month's column—and which will work with both the TRS-80 and the PMC-80—is a special version of KEEPIT (*80* Reviews, December 1980) that resides in 3000 to 33FF. It is sold in a 2716 EPROM package, leaving 3400 to 37DF free for later user programming. The cost of this ROM is under $50.

Personal Microcomputers, Inc., also makes a ROM-only addition for the TRS-80 called REX-80. Like the Memory Sidecar, it is mapped to addresses 3000 to 37DF, but includes no RAM. This can be a savings for those users who do not need read/write memory at these locations. The RES-80 comes assembled and tested for $60, including a zero-insertion-force socket for the ROM. The power supply is $8, if needed, and a standard double-ended 40-pin bus connector is $25.

Finally, a board which installs inside the TRS-80 cabinet and holds either two 2708 1K EPROMs or one 2716 2K EPROM has been created by the Micro 80 Computer Club of Ottawa, Ontario. They call their board the Romplus, and developed it for use by their members. If you would like information, send an international postal coupon for two ounces (and if you wish, a small contribution to the club) to Micro 80 Computer Club of Ottawa, in care of Brian Harron, 67-3691 Albion Road, Ottawa, Ontario K1T 1P2, Canada.



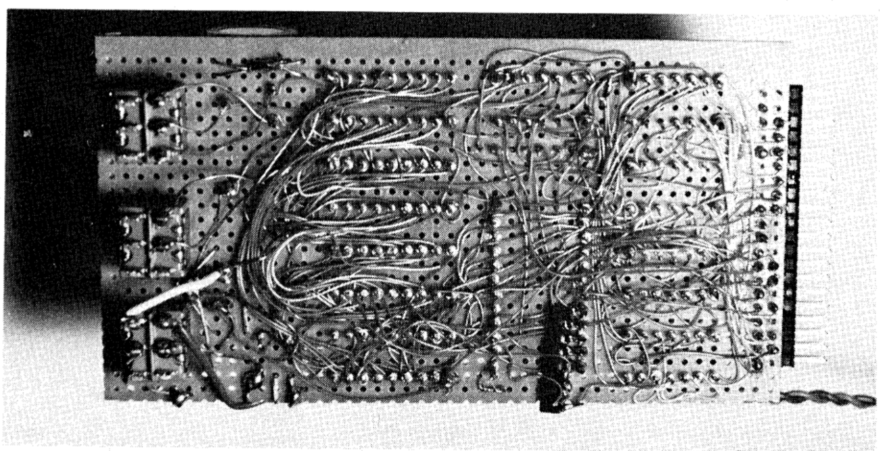
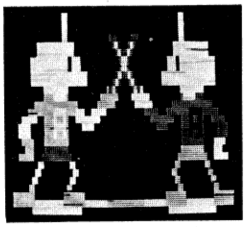


Photo 1-top, Memory Sidecar prototype with 2K ROM and 2K RAM in place, and Photo 2-bottom, Underside showing wire-wrap method of construction.

Get the most from your micro with software and accessories from one of the world's largest selections. ✓17

The Program Store

CALL TOLL FREE 800 424-2738



new

DUEL «N» DROIDS

By Leo Christopherson from Acorn
Your 'droid has already learned NIM, so now it's time to teach it how to wield a laser sword! Leo Christopherson, author of "Android NIM," "Dancing Demon" and other animations, has developed a new type of animation and high-quality sound in his latest work.

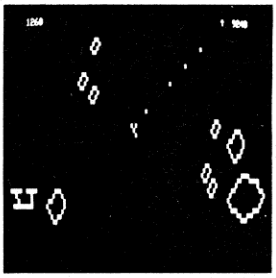
Your 'droid starts out as a lowly clown. You teach it how to use a laser sword by controlling its movements. After training it to be a "Grand Master," you enter the tournament against the program's skilled 'droid! Entertainment for all ages.

Protected Tape...$14.95
Protected Disk...$20.95

DEATH-MAZE 5000

from Med Systems
A new breed of adventuring! Venture through a graphically represented 3-D maze, with halls that could dead end -- or recede to infinity. Step through the doors or drop into the pits. Will you encounter monsters and mayhem, or will you be treated to useful objects and information? Will you ever get out alive?

16K TRS-80, 32K APPLE II...$12.95



SUPER NOVA

By Bill Hague from Big Five
Asteroids surround your ship. You must shoot the asteroids, as well as any alien spaceships. Written in fast machine code, this game is GREAT!

You may encounter five different kinds of alien ships, including the very deadly flagship. You shoot from your ship's position, rotate it, use your thrusters to move -- if you are overwhelmed, you can even get away to hyperspace. Fast and exciting.

Tape....$14.95

DDT Disk Drive Timer

GRAPHIC DISPLAY OF MOTOR SPEED
DRIVE NO.: 0 RPM RANGE: 10
EACH MARK REPRESENTS 0.17 RPM.
(SLOW) (CORRECT) (FAST)
295.00 296.67 298.33 300 301.67 303.33 305.00

from Disco-Tech
Analyze and adjust your disk drive motor speed with a real-time graphic display. Manual details use for Radio Shack, Shugart, MPI, Pertec and Vista drives, and DDT can be used with any drive. All you need is DDT, two screwdrivers and five minutes.

Disk.....$19.95

SPACE WAR

By Device Oriented Games from Acorn
A two-player, real-time action game that lets each player control a spaceship with rotate, thrust, fire, and hyperspace. Five game options (including gravity) and three playing speeds. In fast machine language.

Tape...$9.95

GALACTIC TRILOGY

By Douglas Carlston
Take control of the Galactica as you navigate through an uncharted 3-dimensional universe. In "Galactic Empire," you attempt to unify a universe that is randomly created each time you play.

"Galactic Trader" pits your bartering skills against those of the other inhabitants as you try to accumulate riches and power. But watch out for the assassins and the energy cartel -- they're out to getcha!

Diplomacy and deviousness play equal parts in "Galactic Revolution." It's a game that combines tactics, social manipulation and Machiavellian ruthlessness. For more intrigue, this game allows more than one player. Sound effects.

Choose any game at $14.95 for TRS-80 16K on tape, $24.95 for Apple II 48K Disk.

To control the entire universe, get all three!

JET FIGHTER PILOT

from Instant Software
Launch one of several realistic jet fighters from an airport, or catapult from the deck of an aircraft carrier. Incredibly realistic simulation, right down to maintenance problems.

You will not only learn about the dynamics of flight, you'll discover the complex operation of modern military jet aircraft as you sit back and try to keep up with the constantly changing instrument panel display. Challenging and informative.

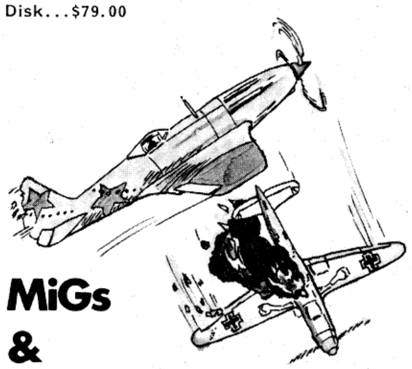
Cassette...$14.95

EDAS Editor/Assembler

By Roy Soltoff from MISOSYS
With EDAS, you are no longer tied to memory limitations while writing in assembly language. Now you can assemble directly from text stored on disk. Branching lets you test your program, then return directly to EDAS. Great for editing and debugging.

Other features include: global editing, upper/lower case support, block moves, plus availability of DOS commands within EDAS. It's the Editor/Assembler designed with the programmer in mind!

Disk...$79.00



MiGs & Messerschmidts

"It is the summer of 1941 and the Blitzkreig is smashing into the heart of Russia..."

This is how your instructions begin when you become the fighter squadron leader in "Migs & Messerschmitts", one of four exciting new Discovery Air Combat Simulations.

These World War II re-enactments are historically accurate -- they challenge you to learn the tactics used by the actual combatants! Written in machine language for fast response.

MIGs and Messerschmidts
RAF: The Battle of Britain
Jagdstaffel
Winged Samurai

For TRS-80, Apple II, PET -- 16K...$19.95

PROGRAMS UNLIMITED...

...if you don't see the program you'd like, give us a call -- we probably have it!

NEWDOS+*........99.95	VTOS*............99.95
NEWDOS/80*.....149.95	RADEX 10*.........99.95
Level III BASIC.49.95	CCA Data Mgr*....75.95
MMS FORTH /Man*.79.95	SYSTEM SAVERS....14.95
KeyEdit.........18.95	Tiny PASCAL*.....50.00

Acorn w/Sound (each): Tape 14.95 Disk 20.95
Pinball Pigskin StarTrek Ting-Tong
Invaders From Space Alien Invasion
Language Teacher (Fr, Span, Ital.)* 19.95 ea.
OSBORNE & ASSOC. Business Systems*: 25.00 ea.
General Ledger Accounts Payable
Accounts Receivable Payroll w/o cost acct.

* Disk

THE PROGRAM STORE · Dept K8 Box 9609 · 4200 Wisconsin Ave, NW · Washington, D.C. 20016

Item ____ Price ____
Postage $1.00
Total ____
☐ CHECK ☐ VISA
☐ MASTERCARD MC Bank # ____ Card # ____ Exp ____
name ____
addr ____
city ____ state ____ zip ____

". . . His students learn BASIC and programming skills as they need them to solve math problems and other assignments."

5th, 6th Graders Practice Logic and Program on High School and College Level

In 1978 Grayson Wheatley loaned his TRS-80 Model I to the Cumberland Elementary School in West Lafayette, IN. It was placed in Jeanne Goris' fifth grade classroom, along with a number of calculators. Though it was not part of a BASIC course, the programming skills and logical thought processes her students have developed are impressive.

Last summer five students from that fifth grade class were among 16 gifted youngsters admitted to a special section of a Purdue University course for computer majors. The accelerated course in Pascal used the standard college text and covered three-quarters of the semester material in 17 days. According to Wheatley, instructor Bill Verts told him, "If I went this fast with Purdue students, I'd lose most of them." The five students from Cumberland Elementary all received As and Bs, and earned their first college credits. This school year they are in the seventh grade.

Cumberland's Gifted

Dr. Wheatley, of Purdue's Gifted Educational Resources Center, and James Hersberger, a Purdue graduate student, began working with Jeanne Goris in 1978 to revise the math curriculum for the top 27 percent of Cumberland's fifth graders. The following year Bob Foerster joined the project and taught the same students in the sixth grade. The school board purchased two more TRS-80s to place in Cumberland. The project, which has been extremely successful, is being continued and extended to the other West Lafayette elementary schools, and copied in Fort Wayne, IN.

At Cumberland Bob Foerster is excited about his students' enthusiasm. Since he has been involved with the program, Foerster believes the students have shown "a tremendous change in attitude in math class." Rather than the fear which most children have of math, Foerster says these students "want to try hard problems, even those without obvious answers!"

Bob Foerster's enthusiasm is shared by many teachers across the country who have the advantage of classroom micros. And like these other teachers, Foerster emphasizes the importance of the computer's problem-solving applications.

". . . Students 'want to try hard problems, even those without obvious answers!' "

(Problem solving has become the popular term among educators for mathematical and applied logic.) Once computers are introduced, most schools teach computer science and include problem solving or heuristics (self-teaching) under that heading. Participants in the Cumberland Elementary project have uniquely reversed the situation.

Dr. Wheatley explains that at Cumberland they are "attempting to teach problem-solving skills, promote curiosity and teach math." Computers and calculators are available as tools towards these ends.

In a report to the West Lafayette School Board, Wheatley and Hersberger outlined the following goals of the cumberland program:

1. Challenge the more able mathematics students.
2. Utilize calculators to develop concepts and build problem-solving skills.
3. Introduce computer programming as a problem-solving tool.
4. Assure computational proficiency.
5. Emphasize the fundamental concepts of mathematics, and develop basic thinking skills.
6. Increase students' intellectual curiosity.

Besides his help in defining goals, Wheatley had a hand in the structure of the project. "Kids are really ripe in fifth and sixth grades" to learn computer skills, he explains. With equal conviction, he adds that children in this age group would find a straight course in programming "dry and boring."

Bob Foerster says that his students learn BASIC and programming skills as they need them to solve math problems and other assignments. Once they begin programming, programming assignments often lead to learning higher mathematics concepts, which in turn lead to more advanced programming. In short, an upward learning spiral is set into motion.

For instance, one boy in Foerster's class decided to write a slot machine program. When he finished the original version, he decided he needed to learn more about probability to improve his program. To further his study of probability, he wrote another program to simulate dice throws. In another case, students studying integer functions and expanded notation decided to write a program to solve their math assignment. They used nested loops in the program, learning that skill as well as their assignment.

Students in the project became prolific programmers, creating a variety of utilities, educational programs and games: several spelling tutorials, many complex target games; a computer dating program that incorporates matrices; a mathematics tutorial used in other Cumberland classes; a drill and quiz of presidents of the United States; a program in which the player manages an American League baseball team in a simulated game; a program to compile and summarize statistics on energy conservation, etc.

Mindless Drill and Practice

While the Cumberland teachers and administrators have been thrilled by the software written by their students, they are discouraged by the courseware (educational software) that is available in the marketplace.

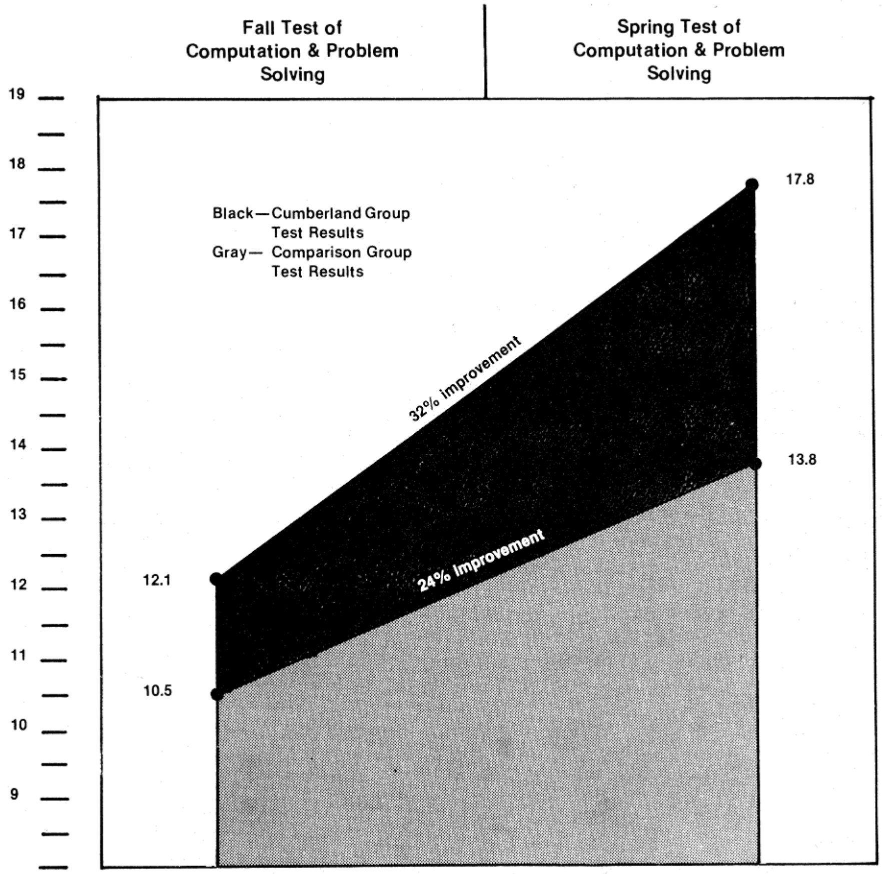
"Most of the software that I see (on the market) is of the mindless drill and practice variety. It doesn't make any sense to use a computer for developing paper and pencil computational skills," Wheatley says. "Computers should be used to develop higher level thinking—reasoning, estimation skills, the application of problem solving heuristics. These are the tools that are needed in a technological society."

Through the Cumberland Elementary School project, Wheatley has had a chance to see his theories, of education for the gifted and of computers in the classroom, in practice. Statistics and observational studies verify the apparent success of the Cumberland experiment.

Studying the Results

The project began after standardized tests were administered to Cumberland's fourth graders. The top 27 percent of those students entered Jeanne Goris' fifth grade. A control group, the top 27 percent of the fifth graders at the other West Lafayette schools, was also determined.

The Cumberland students and the control group were all given a test of computation and problem-solving skills early in the 1978-1979 school year. Another test of these skills was given again towards the end of the year. The Cumberland students began the year 1.6 points ahead of the control group. At the end of the school year, both groups had advanced, but the Cumberland students had pulled ahead of the control group by four points. A graph comparing the test results between the Cumberland class and the other West Lafayette students is shown here.



In reference to these tests, Wheatley and Hersberger reported to the school board: "Testing showed that the students (at Cumberland) learned advanced mathematics concepts compared to their counterparts, and in no way suffered computationally from the reduction in drill and practice time." They also stated that "the fact that nearly every topic studied was extended to levels usually considered appropriate to high school and college students" was another measure of success.

Working with students in Purdue's Pascal course for gifted students, graduate student Tim Fisher did a study comparing the five Cumberland students to five other students in the class. This provides another measure of the project results. In his findings he wrote that the Cumberland group "seems to have better problem-solving processes." He also stated, "They were more enthused, involved and had more fun."

David Flowers, supervisor of Computer Instructional Services for the Fort Wayne, IN school district, spent a day observing Jeanne Goris' fifth grade and Bob Foerster's sixth. His remarks and commentary are filled with accolades. One of several particulars that left a lasting impression on him involved two fifth grade boys who had just decided to write a tennis simulation.

"The typical approach is to start with a reduced version of the problem and then once the simplified version has been developed, modify it until it comes as close to a realistic situation as possible. This is exactly the approach these two boys were taking! As we discussed their plans, they were willing to postpone tackling small details in order to get a simplified version in operation. Many high school students and many of their teachers with whom I've worked were unwilling to tackle simulation problems; the few that did usually had no plan of attack to the problem. These fifth grade students are the exception rather than the rule. They have evidently developed some rather sophisticated techniques for problem solving."

For the most part, Cumberland Elementary School's special curriculum for their gifted math students has met the goals it set at the beginning. However, in their recommendations to the West Lafayette school board, Wheatley and Hersberger pointed out that "there have been students that were not capable or motivated" to meet the challenges. They recommend that only the top 10 to 15 percent of the West Lafayette students participate in the future.

While overall student response does not point to brighter possibilities for the average West Lafayette student, Wheatley and Hersberger contend that the program still has a valid place. The following quote is taken from the concluding remarks of their report to the West Lafayette school board.

"Some may argue that we cannot or should not have a special program for so few. But this argument collapses in light of the athletic program. . . . If these highly able students are not challenged, they may slip into mediocrity when in fact they are capable of much more. The issue is not so much what they learn but their attitude toward learning and ideas." ■

by Nancy Robertson
80 *Staff*

Harvard Micro Conference: Education for Publishers and Teachers

The irony is perfect. A serious information gap surrounds the introduction and use of the machines that have created the information explosion. Perfect though it may be, this irony is unacceptable for many within the educational community.

The painful slowness inherent in today's parallel evolution of techniques for implementing microcomputers in the classroom is particularly frustrating considering the speed with which micros are being woven into the fabric of our lives. For contemporary educators trying to bring computers into their classrooms, it is usually a case of the right hand not knowing what the left has been up to. The paucity of journals covering the topic, the institutions researching the subject, and funding sources that bankroll the projects all confound an already perplexing situation.

> *"Schwartz . . . urged his audience to think beyond the drill-and-practice mindset . . . microcomputers have come to symbolize."*

A November conference on the use of microcomputers in the classroom, sponsored by Harvard University's Graduate School of Education, was an attempt to breach the information gap that the information machines created. The Harvard conference reflects current trends in education regarding the use of computers in the classroom. At Harvard's Gutman Library, approximately 500 teachers, administrators and researchers gathered for three days to discuss theories, swap ideas, tell tales and share enthusiasm. The conference also attracted the attention of manufacturers who sense the potential of the burgeoning educational market.

For most of those in attendance, the emphasis was not on using the computer to teach computer science. Instead, the development of the computer as another tool that creative teachers can employ to make learning meaningful and fun was stressed. Though the appropriateness of the micro in drill and practice environments was not minimized, most conference attendees were looking beyond this application toward more innovative uses of the machines.

Theory and Practice

The carefully structured conference was a healthy mix of the theoritical and the practical. Keynote speaker Judah Schwartz, an MIT professor, researcher of computer simulation of human intuition, and theoritical physicist, stressed the use of the machine in new and different ways. He urged his audience to think beyond the drill-and-practice mindset that classroom microcomputers have come to symbolize. In his keynote address, Schwartz maintained that the classroom computer's most important task is to function as an intellectual amplifier for the student.

An extensive offering of specialized seminars in which educators detailed their experiences with kids and computers was available throughout the conference. Topics included: K through 8 math drill with interactive video disk, long and short range planning for computer use in schools, software development, the LOGO experiments, microcomputer selection, microcomputers at the community college level, public access computers and much more.

Radio Shack's Chris Bowman at Harvard

Publishers' Panel

A publishers' panel, composed of representatives of many large text book publishing houses, discussed the problems they have encountered in the development of quality, educational software. The panel, chaired by Christopher Cerf (son of Random House and "What's My Line" veteran Bennett), included representatives from McGraw Hill, Houghton Mifflin, Milliken, Ginn & Company and Bell & Howell.

It is not surprising that these textbook publishers intend to go into the software business. The surprise was, that in terms of ideas and abilities, these corporate giants have no monopoly on expertise when compared with teachers who are already writing and using educational software. Everyone, it seems, is starting at square one.

In the realm of educational software, as in many other areas, microcomputers are altering traditional political and power relationships. The aggressive tone of the audience's questions indicated that teachers will demand a larger role than they have in the past in deciding what is—and is not—appropriate for use in their classrooms.

The publishers emphasis on ROM-based, non-modifiable, drill and practice software did not strike a responsive chord among the educators present. If publishers persist in their efforts to market software of this type, they may find themselves short on sales and long on inventories as teachers develop and use their own programs.

> *". . . Corporate giants have no monopoly on expertise. . . ."*

The Leading Edge

If textbook publishers are on the trailing edge of the classroom computer revolution, MIT researcher Seymour Papert is on

Continues to page 56

$ DISCOUNT $
TRS-80® DEALER

MODEL II

26-4002
64K 1 Drive
$3440.00

26-4160 1 Drive EXP	$1035.00
26-4161 2 Drive EXP	1575.00
26-4162 3 Drive EXP	2115.00
26-4501 Gen. Ledger	180.00
26-4502 Inventory	180.00
26-4503 Payroll	360.00
26-4554 Acct. Rec.	180.00
26-4701 Fortran	270.00
26-1157A Daisy Wheel	2495.00
26-1158 Daisy Wheel II	1799.00

COMPUTER SPECIALISTS

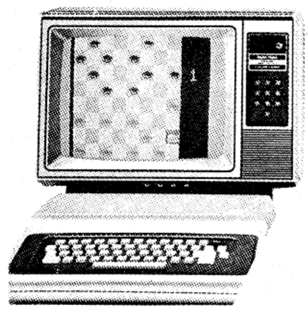
26-1155 Quick Printer II	$187.00
26-1145 RS-232 Board	84.00
26-1140 "O" K Interface	249.00
26-1141 "16" K Interface	359.00
26-1142 "32" K Interface	469.00
26-1160 Mini Disk - Drive O	419.00
26-1161 Mini Disk - Additional	419.00
26-1154 Lineprinter II	699.00
26-1156 Lineprinter III	1799.00
26-1159 Lineprinter IV	859.00
26-1166 Line Printer VI	1080.00
26-1563 Scripsit - Disk	79.00
26-1566 Visicalc	83.00
26-1562 Profile	72.00

NOTE: Call for availability of VIDEO TEX, Model III, Color, and other new products.

MODEL III

26-1061 4K I	$630.00
26-1062 16K III	888.00
26-1063 32K III 2-Drives, RS232	2225.00

COLOR



26-3001 4K	$360.00
26-3002 16K	540.00
26-3010 Color Video	360.00
26-1206 Recorder	54.00
26-3008 Joysticks	22.50

CENTRONICS

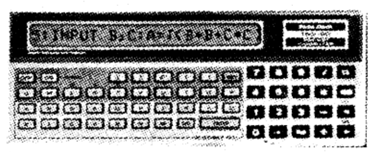
Fast 100 CPS Centronics 730 Printer	$659.00
Text Quality Centronics 737 Printer	$819.00

Model II Cobol Compiler
$360.00
Cobol Run Time Package
$36.00

ALL OTHER R.S. SOFTWARE FURNITURE, STANDS, CABLES AND ACCESSORIES AT DISCOUNT FROM CATALOG PRICE.

Novation Cat Modem	$149.00
CCA Data Management System	72.00
Adventure Games Games 1-9 each	14.00

Pocket Computer



26-3501 1.9K P.C.	$225.00
26-3503 Cassette I/F	45.00
14-812 Recorder	72.00

Acorn Software Products, Inc.

GAMES:	
Alien Invasion	$9.00
Stock Market	9.00
Star Trek	9.00
Block 'Em	9.00
Ting-Tong	9.00
UTILITIES:	
System Savers	14.00
EDUCATION:	
Language Teacher	18.00

FREE: COMPUTER CATALOG UPON REQUEST

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC. ✓72

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No.

R.S. 90 Day Limited Warranty
F-48 Form Provided

Largest Inventory In the S.E. U.S.A.

*TRS-80 is a registered trademark of the Tandy Corp.

✓Reader Service—see page 242

Random House Markets Radio Shack's Line

Continuing in its assault on the educational market, Radio Shack has named Random House, Inc. an authorized educational distributor for Tandy products. Random House, in addition to being a major publisher of books, is also a widely recognized distributor of classroom materials.

Word of the new agreement came from Charles A. Phillips, senior vice-president in charge of special markets for Radio Shack. Phillips termed the contract "an important step in Radio Shack's strategy to better address the growing market (for microcomputers)... in the schools." Random House will now carry all of Radio Shack's products—both hardware and software—through its extensive institutional marketing network.

For Tandy, the timing of the team-up seems particularly propitious. February 1981 marks the second anniversary of the firm's educational sales program; Tandy now has fielded a five-man team of educational sales coordinators who work expressly with teachers and school administrators within their respective regions (see *Microcomputing Industry*, Jan., 81 and *80 Microcomputing*, Jan., 81). During this time, Radio Shack has secured a large piece of the low-end micro market within academia, a market which it codominates with Commodore and Apple. The alliance with Random House promises to greatly enhance Radio Shack's position.

Talking with Teachers

As to whether this new coalition will siphon off significant business from Radio Shack's own retail outlets, Phillips thinks not. "We won't be stepping on our own toes," he said, "there's really no conflict there. For one thing, of the people manning our stores, only a small percentage talk with teachers. Also, our retail people generally lack the patience to make educational sales which usually require months to complete. However, we do plan to structure some sort of reward system for our retailers, which will take into account outside-store sales to schools within their respective territories."

Radio Shack currently offers about one dozen courseware programs. Random House will be adding these to its own inventory. "Math-based programs will be an easy entree for us," continued Phillips, "from there we can readily branch out into other subject areas such as those that rely heavily on graphics."

The educational market is usually broken down into three broad categories: computer-aided instruction (CAI), the study of computers themselves, and administrative or non-classroom applications. Random House and Tandy will now be tackling all three in tandem and, in addition to the expected rise in revenues, both firms hope to reap side benefits in the form of new courseware. These programs are scheduled for co-development through an ongoing R&D program conducted within various academic environments.

New Directions

More interesting than the actual mechanics of the new Tandy-Random House agreement, however, is the fact that for the the first time Fort Worth has decided to deviate from its traditional policy of strictly in-house marketing and enter into a contract with a gentile company. Nor will this be the last such tangent for Tandy as can be readily inferred from Phillips' remark that the Random House agreement is nonexclusive and that at least one other similar arrangement is being negotiated with another educational publisher.

It is reasonable to assume that the markedly different character of institutional sales led Tandy, at least in part, to find outside talent, this to gird its loins for the market wars to come. ■

by Paul Quinn
80 Staff

Harvard Conference

Continued from page 54

the leading edge. Papert's seminar, discussing his 10 years of experimentation with young children and computers, detailed his efforts to create a transparent man/machine interface. His experiments have resulted in a new, symbol oriented language called LOGO. LOGO allows children to interact with computers despite the fact that the children cannot read. As interaction occurs, the children are learning how to learn.

Papert's pioneering work is setting the stage for educational computing in the next decade. He predicts that if the correct decisions are made now, and the inherent conservatism of educators can be overcome, the attainment of computer literacy in children will be as natural as the attainment of language skills and will occur without deliberate instruction.

Papert places heavy emphasis on the role that the interactive graphics and symbol rich environments of microcomputers can play in the early development of cognative skills. He even thinks that our society can overcome its inherent mathophobia with the help of computers.

Using learning aids like his Turtle, a cybernetic, student-programmed robot which traces geometric shapes on paper, students will no longer find math a foreign domain because they will actively participate in the creation of geometries, sets, etc. They will do so by programming their Turtles to draw. In the process, they will also acquire a logical (top down) approach to problem solving.

A self-described computer utopian, Papert believes that the world will be a better place thanks to computers. And, contrary to popular opinion, Papert feels that the computer will intensify our personal relationships, especially with our children, as it hastens their intellectual development.

Cross-Pollination

As participants filtered back to their districts, classrooms and projects all across America, the result of the cross-pollination that occurred at Harvard will begin to take effect. At the very least, the people from New York who've been working on a long range plan for the instructional uses of microcomputers in grades K through 8, now know what has been going on in Palo Alto, where studies are underway to determine exactly what it is that makes computer games fun. And, Rosemount, Minnesota's experience with interactive video disks is now common knowledge in Dallas where they, too, are experimenting with computer controlled, multimedia instruction.

As the halting, first steps of computerization of the classroom are taken, gatherings like Harvard's will provide the energy and direction for future growth. Next year's conference is already being planned.

Inquiries should be directed to the Media Center, Monroe Gutman Library, Harvard University Graduate School of Education, 6 Appian Way, Cambridge, MA 02138. (Telephone 617-495-4225.) ■

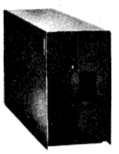
by Chris Brown
80 Staff

Toll-free order no.:
1-800-527-4196†

LLOOK!!

TRS-80*Owners Save on Equipment & Software!

ACCESS Mini-disk Systems



Access Unlimited's own economy mini-disk systems store more data, are more reliable. Data access times are fastest possible with your Expansion Interface. Heavy duty power supplies run cooler, last longer. Low noise three-wire ac power cord is safer. Enclosures are finished in compatible silver enamel.

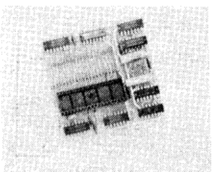
AFD-100ᵀ (40-track, 102 Kbytes) $315.00

Mention our DOUBLE DISCOUNT NUMBER when you order and save $20.00 on your AFD-100!

Percom Mini-disk Systems

TFD-100™ (40-track, 102 Kbytes/side) $349.95
TFD-200™ (77-track, 197 Kbytes) $634.95
Prices include Percom upgrade PATCHPAK™

DATA SEPARATOR™



This PC board plug-in adapter for the TRS-80* virtually eliminates data read errors (CRC error — Track locked out!) which occur on high density inner disk tracks, a problem that has plagued TRS 80* systems. The Percom Data Separator™ is installed in the Expansion Interface without modifying the host system. Caution: Opening the TRS-80* Expansion Interface may void the limited 90-day warranty: $29.95.

Percom OS-80™

An advanced easy-to-use disk operating system that works with Level II BASIC commands. Resides in only 7-Kbyte of memory. May be extended indefinitely with disk-resident utilities. Supplied on 5″ disk with example programs: $29.95 with instructions.

CIRCLE J Software

Two extremely useful utilities for Percom's OS-80™ DOS:

1. Machine Language Save/Load Utility. On 5″ disk with bonus patch program that allows RS Renumber Utility to run under OS-80™. $14.95, with instructions.
2. VARKEEP — Adds NAME SAVE and NAME KEEP commands to OS-80™. Use one set of common data with two or more BASIC programs. Also runs under Radio Shack DOS. On 5″ disk, with instructions: $14.95.

Z80ZAP

Super fast machine language disk modification utility. Read, Write, Display, and Modify sectors; remove passwords; apply patches, fixes; make backups and much more. On 5″ disk with instructions: $29.95.

Ask about Scott Adams' Adventure games!

100% machine language word processor . . . SPECIAL DELIVERY (From Software Etc.)

Use MAILFORM to create name and address lists; EXTRACT to find names by ZIP, address, gender, age, etc.; SORT to sort an entire list on any field in seconds. Print personalized letters written with either the Electric Pencil‡ or Scripsit* using MAILRITE. Prints labels from Mailfile created under MAILFORM. Runs under Percom's OS-80™, Radio Shack's TRSDOS*. $125 (disk)

Inexpensive Color Graphics: the Percom Electric Crayon™

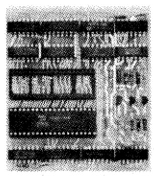
Spectacular multicolor graphics, sharp 2-color alphanumerics with your TRS-80*, a color tv and the Percom Electric Crayon™. Up to eight colors. Resolution with full display memory (6 Kbytes) is 256 X 192 picture elements. Microprocessor controlled the Electric Crayon™ is not only a full color graphics system but also a complete, self-contained control computer with a dual bidirectional parallel I/O port — provision for second dual port. Interface the TRS-80* via your Expansion Interface or Printer Cable Adapter. Supplied with 1 Kbyte display RAM, EGOS™ operating system and comprehensive users manual with example programs. $249.95.
Optional TRS-80* interconnecting cable: $24.95.

Percom's Speak-2-Me-2™

Give your TRS-80* the gift of speech

Texas Instruments' Speak & Spell~ is the voice of your TRS-80* computer with this clever interface module manufactured by Percom. Your own Level II BASIC programs announce, command, implore with sentences and expressions formed from Speak & Spell's~ vocabulary. The Speak-2-Me-2™ PC module installs in the battery compartment of your Speak & Spell~. Power is supplied from an ordinary calculator power pak. Comes with interconnecting cable (for TRS-80* EI or Printer Cable Adapter), operating software and users manual: $69.95. (Speak & Spell~ not included.)

the DOUBLER™



Percom's new plug-in adapter for your Expansion Interface stores almost twice the data on a diskette track as a single-density system. You can store up to four times more data — depending on the type of drive — on one side of a diskette than you can store using a standard Model I mini-disk drive. Other features: Reads, writes and formats *either* single or double density minidiskettes. • Runs TRSDOS*, NEWDOS +, Percom OS-80™ or other single density software *without* changing either software or hardware. Switch to double-density when convenient. • Includes DBLDOS™ a TRSDOS* compatible double-density operating system. • Includes on card, high-performance data separator circuit. • Installs *without* rewiring or trace cutting. • Introductory price, including DBLDOS and format conversion utility, only $219.95. Mention our DOUBLE DISCOUNT NUMBER when you order and save $20.00 on your DOUBLER!

DOUBLE-ZAP — Modifies NEWDOS 80 + for double-density operation using the Percom DOUBLER. Permits simultaneous single- or double-density operation. From Software Etc. and Circle J Software. On minidiskette. $49.95

Use your credit card and save!

VISA and MasterCard charges are not deposited until the day your order is shipped.

Disk System Interconnecting Cables

Improvement over RS cable design places drive 0, which includes the cable termination, at the end of the cable to eliminate the reflected noise of an unterminated cable. Better data integrity. Prices:

Two-Drive Cable $24.95
Four-Drive Cable 34.95

Power Line Filter

115/250 V, 50-400 Hz. Instructions included for easy installation in standard mini-box chassis: $19.95

Minidiskettes

10 Disks in a convenient plastic organizer box $34.90
Single Disk 3.49

Disk Drive ID Tabs

1″ x 1-¼″ self-adhering plastic drive identification tabs. Compatible silver with engraved black drive number. Two tabs (Nos. 0, 1): $2.50; three tabs (Nos. 0, 1, 2): $3.25; four tabs (0, 1, 2, 3): $4.50.

How to Order

Order by calling Access Unlimited toll-free on 1-800-527-4196†.

Mail orders also accepted. Orders may be charged to a VISA or Master Charge account, or paid by a cashier's check, certified check or money order. We accept COD orders with 25% deposit. Sorry, we cannot accept personal checks. We pay shipping and insurance charges on orders over $1,000.00. Add approximate insurance and shipping charges for under $1,000.00. If in doubt about these charges, ask when you call in your order. Texas residents include 5% sales tax. Minimum order: $20.00. Allow 2 to 4 weeks for delivery. **†Texas residents call (214) 494-0206**

ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE AND ALL OFFERS SUBJECT TO WITHDRAWAL WITHOUT NOTICE.

DOUBLE DISCOUNT NUMBER: 80M110

ACCESS UNLIMITED
315 N. Shiloh · Ste. D1 · Garland, TX 75042
(214) 494-0206

† trademark of Apparat Corporation.
ᵀ trademark of Access Unlimited Company.
™ trademark of Percom Data Company, Inc.
* RADIO SHACK and TRS-80 are trademarks of Tandy Corporation
~ trademark of Texas Instruments Corporation
‡ trademark of Michael Shrayer Software, Inc.

Combine accurate flight characteristics with the best in animation graphics and you'll have SubLOGIC's

T80-FS1 Flight Simulator

for the TRS-80

SubLOGIC's T80-FS1 is the smooth, realistic simulator that gives you a real-time, 3-D, out-of-the-cockpit view of flight.

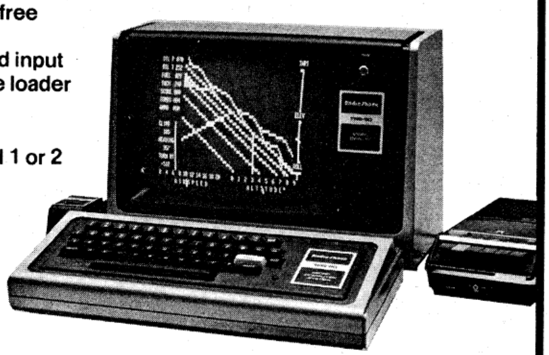
Thanks to fast animation and accurate representation of flight, the non-pilot can now learn basic flight control, including take-offs and landings! And experienced pilots will recognize how thoroughly they can explore the aircraft's characteristics.

Once you've acquired flight proficiency, you can engage in the exciting British Ace 3-D Aerial Battle Game included in the package. Destroy the enemy's fuel depot while evading enemy fighters.

Computer and aviation experts call the T80-FS1 a marvel of modern technology. *You'll* simply call it *fantastic!*

Special Features:
- 3 frame-per-second flicker free animation
- Maximum transfer keyboard input
- Constant feedback cassette loader

Hardware Requirements:
- Radio Shack TRS-80, Level 1 or 2
- 16K memory
- *Nothing else!*



Only $25

See your dealer or order direct. For direct order, include $1.25 and specify UPS or first class mail. Illinois residents add 5% sales tax. Visa and Mastercard accepted.

subLOGIC
Distribution Corp.
Box V, Savoy, IL 61874
(217) 359-8482

✓150

NEW GIN — NEW CRIBBAGE

FOR MODELS I AND III AND ATARI 800

GIN RUMMY 3.0 The classic computer Gin Rummy is now even better, with card graphics and faster playing time. On-screen cards can be rearranged as you play, and the program plays the same mean game of Gin, holding its own against anyone. Plays a full regulation game and changes strategy to counter your play. Can you beat Gin Rummy 3.0? TGR-2 $16.95

ATARI GIN with color graphics and sound for Atari 800, 32K. IGR-1 $19.95

CRIBBAGE MASTER plays a strong game, too, making the most of every play, hand and crib. It'll Muggins you for the smallest mistake, but try to catch it counting wrong. Card graphics. TCM-1 $12.95

CONCENTRATION Clever screen graphics plus sound effects make this an irresistible computer version of the long-popular match-the-cards game. Play with 2 to 15 pairs of figures — the smaller game is ideal for children, and 15 pairs will challenge anyone. TCO-1 $9.95

LABYRINTH RUN Race through sharp turns, slaloms and narrowing passages. A fascinating/frustrating test of coordination, with three skill levels. High speed graphics. TLR-1 $9.95

E.S.P. LAB Psychic? Find out with this program based on the famous Duke University experiments. See the full-page review in the October issue of Kilobaud. TPL-1 $9.95

THE LISTMAKER II Pull any category from a list in seconds with this powerful, versatile program. Enter up to 400 names or items, with codes, in 16K. Sort, edit, dump, load, print or display lists and sub-lists on screen. TLM-2 $9.95

CALCULATOR PLUS makes your TRS-80 a printing calculator. Chain and mixed calculations, constants, memories. On-screen review of long add-and-subtract operations. Prints all calculations. TPC-1 $9.95

CHECKBOOK PLUS Never agonize over a bank statement again! Put the figures in and let Checkbook Plus handle all the details and find the errors. Special check-stub arithmetic review. TCB-1 $9.95

CALCULATOR & CHECKBOOK both on one cassette. TCC-1 $14.95

California residents add 6% sales tax.

VISA · MasterCard

24-hour credit card order hotline: (213) 454-8290

MANHATTAN SOFTWARE ✓90

Post Office Box 35, Pacific Palisades, CA 90272

TRS-80™ SOFTWARE

For Mod I & Mod III

MACHINE LANGUAGE SOFTWARE

MONITOR #3 .... $39.95
Disassembler; memory displays; memory move, search verify, and modify; read and write object tapes; hexadecimal arithmetic; object code relocator; unload programs for disk; symbolic output tapes; 41-page instruction manual.

MONITOR #4 .... $49.95
Same as Monitor #3 but adds: save and read disk files; direct input and output of disk sectors; send, receive, or talk to another computer via RS-232-C interface; symbolic disassembly on disk.

SMART TERMINAL $49.95
Enables your TRS-80 to be used as a remote terminal to a time-sharing system. Supports lowercase and full range of control keys. *Automatic transmission* between memory and host computer. Much more.

FASTSORT $9.95
Machine-language sorting program for use by Basic programs. Many times faster than other methods!

GAME OF LIFE $5.95
John Conway's game of "life" shows patterns evolving and changing swiftly before your eyes. A dazzling demonstration program!

BASIC SOFTWARE

MAILING LIST $69.95
Maintains mailing list files of over 1000 names per diskette. Add, delete, change, find name, machine language sort, print file.

SMALL BUSINESS ACCOUNTING $49.95
Based on Dome Bookkeeping Journal #612, keeps track of income, expenditures, and payroll for a small business of up to 16 employees. Daily, monthly, year-to-date summaries.

HOME BUDGET $49.95
Checkbook maintenance combined with records of income and monthly bills. Monthly and year-to-date summaries showing tax deductions.

DATABASE MANAGEMENT $29.95
Defines files of any description and maintain on cassette or disk. Add, change, delete, find, sort, justify, print, line print, total fields, write.

HOWE SOFTWARE ✓103
14 Lexington Road
New City, New York 10956

(*) TRS-80 is a registered trademark of Tandy corp.

History, Music, Math—Micros in Every Class

There's something new and exciting happening in education and computers. There's a public school in North Carolina which is using micros as an integral part of their entire educational program. The school is called the North Carolina School of Science and Mathematics. Located in Durham, NC, it's new this year. It is a coeducational residential school designed for eleventh and twelfth graders from around North Carolina who are considered gifted in science and mathematics. It is on these subjects that the school places its emphasis, and computers are a large part of that emphasis.

According to Dr. Steve Davis, head of the Computer Science and Math Department at the school, it's important to instill a sense of the computer as a tool in all areas of learning, not only in the traditional areas of math and science. To this end, many of the students' classes will be taught as interdisciplinary studies, and the computer will be used in classes such as music, American Studies, and others.

The school, at this time, during its first six months, consists of 150 high school juniors and 13 to 14 full time teachers. About 60 percent of the teachers hold Ph.D.s in their fields. They come from the varied worlds of research, college professorships and high school teaching. Only a few are actually competent on the computers. In fact, some of the students have much more skill in BASIC than their teachers. Teachers do not find this situation a problem, but rather it fits nicely into the philosophy of education and computers that they are trying hard to develop.

The main emphasis this year in computer education is on micros. The school has three Apples and one TRS-80. They expect to acquire two more Apples in the near future, and even more Apples in the coming years.

Hardware and Software

The 80 which they now have is a 32K Model I; unfortunately, it hasn't worked since they got it. The school had hoped to provide the varied experience of different micros, and find the lack of an 80 frustrating. (The local dealers and repair shops haven't been able to fix it; we'll see if the people in Ft. Worth can solve the problem.)

The educators prefer the Apple because of its high resolution graphics, which are unavailable on the 80. Graphics appeal strongly to the students.

The school is hooked into the North Carolina Educational Computer Service, which is a computer network provided by the Triangle Universities (Duke, Carolina University and North Carolina State). The network serves as an umbrella for other campuses in the state and currently services about 25 organizations. The North Carolina School of Science and Mathematics uses the net for its interactive BASIC, Pascal compiler, FORTRAN, and other services.

The School of Science and Mathematics also plans to purchase a VAX 11/750, a product of Digital Equipment Corp., next year. The VAX will offer greater graphics abilities and virtual memory. Students will use the 32-bit CPU VAX with Pascal and C to learn to operate with these computer languages. The VAX will also allow teachers in the foreign languages, American studies, and other disciplines to use the computer for drill, simulations, and computer assisted instruction. The micros currently used are better suited for science, music and math, it is explained by Davis.

Educators at the school are, on the whole, unimpressed with available educational software, and will do without, or create their own. The Apple Education Foundation has donated equipment to further this goal, retaining distribution rights on any software produced. Although the school is not in the software business, they see one of their major goals to be the creation of a philosophy and software that can be utilized in schools throughout the state and nation, Davis said. They hope to develop this software in such a way that teachers with no training in computers will be able to take advantage of the programs.

Curriculum

Using the available hardware, students at the School of Science and Mathematics are learning BASIC and Pascal in independent studies this year. As the languages are mastered, these high school juniors will be writing programs involving simulations, graphics, adventure games, and programs to facilitate the school's administration. In science and math courses, the computers will be used as a lab tool to process data, much as they would be used in a professional research situation.

The computer rooms are open for student use until 10 p.m. each evening, permitting ample time and opportunity for students to graduate with a knowledge of computer languages, programming skills, and many accumulated computer hours.

Davis says that micros are cheap enough for any school system to use—and provide benefits that outweigh the capital outlay. He hopes the School of Science and Mathematics will forge a path for others to follow. Davis believes the school is in a unique position to institute the computer as an integral part of the high school educational process. The teachers and students are highly receptive to computers as an educational tool within an interdisciplinary curriculum. Davis feels these avenues should be opened to students of all capabilities.

In an effort to spread the word and share their philosophy of education, the North Carolina School of Science and Mathematics expects to open its doors to students from outside North Carolina in the future. This summer it will begin a study program by invitation for teachers around the state. Two workshops on the uses of microcomputers, elementary programming and computer applications in math and science are planned for 120 teachers for the first summer program. ■

by Debra Marshall
80 Staff

Boston Wrap-up

In terms of brute square feet and numbers of exhibitors, the Northeast Computer Show was a giant. Larger than either the Chicago or New York shows, Boston attracted over 52,000 computer buffs. Playing to the second largest computer marketplace in America (San Francisco being the largest), Boston drew a wide cross section of exhibitors. Representing the corporate giants were IBM, Prime, Nixdorf, Wang, Bell Telephone, etc. Micro companies were also there in force with Tandy and Apple enjoying the highest visibility.

Radio Shack's tasteful and functionally designed walk-thru exhibit had at least one of everything including a dual disk drive Model III, the new daisy wheel printer, Scripsit for the Model II, and an updated version of Model II TRSDOS. While the Shack's sales staff got down to serious business with the crowds of the curious and the moneyed, there was another Radio Shack booth in back. Here high school kids stood in line for a chance to write their own programs to printout on a daisy wheel.

The noon crowd on Saturday made buying a hot dog or buying a light pen equally difficult. ■

Electronic Dictionaries Will Even Spell

Daddy, how do you spell 'suede'?"
"Go look it up in the dictionary."

The child runs her fingers over a keyboard, slowly picking out the letters S-W-A-D-E. The screen displays, " 'Suede' is a kind of leather. 'Swayed' is a form of the verb 'to sway', meaning to rock or swing back and forth. Please touch your finger to the word you wish."

Dictionaries are with us more today than ever before, and in forms unheard of even 20 years ago. Small books and "unabridged" ones, dictionaries of slang and of acronyms, thesaurus-style volumes and concordances, specialized technical dictionaries, even dictionaries of ethnic and regional variants. The presence of these has, in sum, mirrored our culture's comings and goings, developments and transformations.

Although dictionaries have changed form many times since their introduction as sociopolitical encyclopedias four centuries ago, a new government-sponsored report suggests they are about to take a quantum leap into the realm of electronic information.

A new report from Carnegie-Mellon University predicts the development of a completely computerized dictionary. Preliminary results of a study on such computerized wordstores, commissioned by the National Institute of Education, was published by Mark S. Fox, Donald J. Bebel, and Alice C. Parker in *Computer*, July, 1980.

The actual costs of computer hardware have come down so far, say the authors of "The Automated Dictionary," that portable electronic dictionaries for students will soon be possible. "Predicted storage technology enhancements will make a portable device with 30,000 words feasible in the late 1980's," and video disks will allow enormous storage at low cost. Moreover, these dictionaries are likely to be more useful than printed ones, suggest the researchers.

Main Difficulties

Presently the main difficulties facing creators of an automated dictionary—aside from the significant public relations aspect—are the method of access and the amount of information stored. Saving a list of words is no problem; but given the extensive definitions, pronunciation, examples, forms of speech, etymology and cross-referencing of the average printed dictionary entry, the concept of worthwhile automation comes into question.

If a parent's command to "look it up in the dictionary" were only to help a child learn spelling, then the process might almost be trivial. Beyond, that, "leafing through" the entries has always been part of vocabulary building. Such leafing requires saving and presenting pages of information in a highly interactive and user-oriented manner. But rapidly providing a screenful of dictionary entries complete with synonyms and antonyms demands high-speed search capabilities and enormous storage.

The authors of the Carnegie-Mellon report have concentrated on the educational aspect of dictionaries, but a recent advertisement for Spellguard from Innovative Software Applications in Menlo Park, CA presents another possible use for the automated dictionary: computer as proofreader. For $295, the user of a CP/M based system can obtain a 20,000-word dictionary on diskette. It is compatible with the most popular word processing systems, including Electric Pencil, and can be expanded and customized to include words for particular fields such as medicine and computing.

Likewise, text editors have been making their way into business offices and into the hands of writers—as well as into the newsroom. When asked her opinion of the concept of an automated dictionary, one city desk editor, who currently uses a text editor to file all stories, responded immediately with, "Great! Where can we get one? Can we put it on our system?"

The Carnegie-Mellon report, together with the Spellguard advertisement, suggest that the time may be right to begin creating extensive electronic dictionaries.

But the public relations aspects of a computerized dictionary will not be easy. In a recent article, national syndicated columnist Neal Pierce lets his imagination wander: "What if electronics had come first?" he asks. The difference would be astounding, he suggests, if our latest discovery were "the simple notion of applying ink to paper to create a phenomenon known as the printed page."

"Instead of getting your news from talking heads or bleary little green letters illuminating a screen of limited space, you could have an entire news and entertainment package—it might be called a newspaper—delivered at your door.... In a random, curious way, you could leaf through the sections...."

Individual Dictionaries

The Carnegie report authors recognize this challenge by offering the vision of dictionaries tailored to the individual: "What information is chosen for inclusion in the automated dictionary is dependent upon many variables, such as the type of user (child vs. adult, writer vs. physicist) and storage availability (such as 3000 complete entries vs. 10,000 abbreviated entries). An automated dictionary can be designed as a specialized dictionary, the dictionary can be an easily replaced memory module, or it can have a network connection for changing the dictionary. Alternatively, the data base can be stratified to present difficult information to different users, according to user profile or command."

The authors contend that acceptance depends on the user, and they name seven immediate factors affecting that perception: time spent using it, user error, novice learning, functionality, ease of recalling its use, concentration, and fatigue. Beyond that, they suggest that in the educational environment other factors, including variety, curiousity, fun and adaptability, come into play.

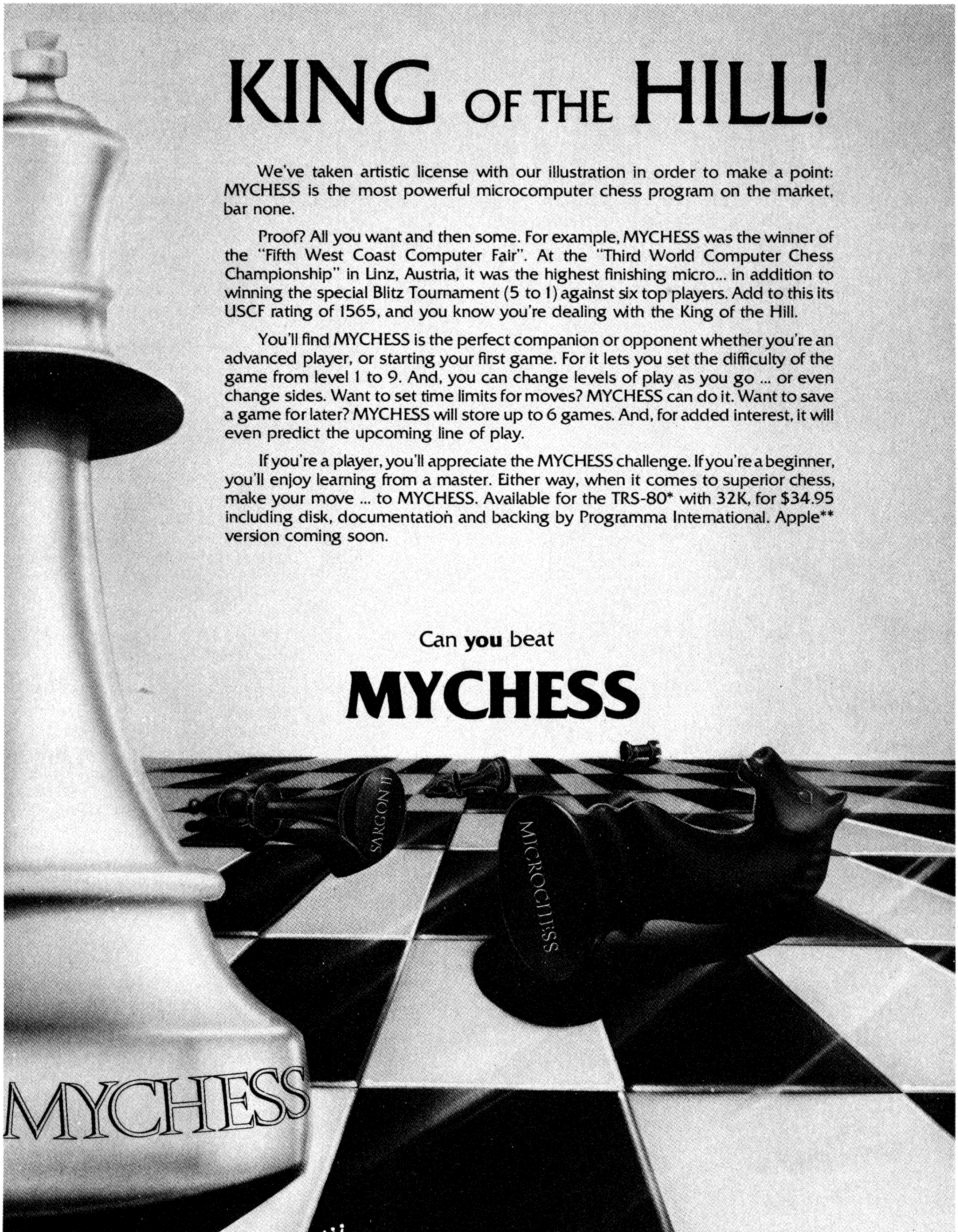
When presented with the idea of an electronic dictionary, teachers and writers are often skeptical. One writer believes that such a tool will make students lazy and wordsmiths bored, and that the cold, electronics medium will detract from the surprise of discovering new words and new meanings.

Etymology

Poet John Ciardi of Princeton has written often about the "ghosts" of words, the long-forgotten history of the words we use that influence the way they are heard. But automated dictionaries, suggest the authors of the Carnegie report, can be searched for any combination of factors—not only meanings, or rhymes, or spellings, but also etymology. That means that the reader need not know Latin (or have unending patience) to discover that fertile, latitude and tolerate have the same ancestral ghosts. Or that speaking about the "essence of the future" is evoking ghosts of the verb "to be."

The authors state it less poetically: "The basic philosophy of Zog (an experimental electronic dictionary system) is that a menu-selection system can be an effective communication method if the user can move around in the system quickly and if there is a large network available to meet the user's needs. . . . Different paths would be provided for different levels of users." ■

by Dennis Kitsz
80 *Staff*



KING OF THE HILL!
We've taken artistic license with our illustration in order to make a point: MYCHESS is the most powerful microcomputer chess program on the market, bar none.
Proof? All you want and then some. For example, MYCHESS was the winner of the "Fifth West Coast Computer Fair". At the "Third World Computer Chess Championship" in Linz, Austria, it was the highest finishing micro... in addition to winning the special Blitz Tournament (5 to 1) against six top players. Add to this its USCF rating of 1565, and you know you're dealing with the King of the Hill.
You'll find MYCHESS is the perfect companion or opponent whether you're an advanced player, or starting your first game. For it lets you set the difficulty of the game from level 1 to 9. And, you can change levels of play as you go ... or even change sides. Want to set time limits for moves? MYCHESS can do it. Want to save a game for later? MYCHESS will store up to 6 games. And, for added interest, it will even predict the upcoming line of play.
If you're a player, you'll appreciate the MYCHESS challenge. If you're a beginner, you'll enjoy learning from a master. Either way, when it comes to superior chess, make your move ... to MYCHESS. Available for the TRS-80* with 32K, for $34.95 including disk, documentation and backing by Programma International. Apple** version coming soon.
Can you beat
MYCHESS
SARGON II
MICROCHESS
MYCHESS
PROGRAMMA
3400 Wilshire Blvd.
Los Angeles, Ca 90010 (213) 384-0579
*TRS-80, a Tandy Corp. trademark. • Microchess, a Personal Software, Inc. trademark.
Sargon, a Hayden Book Co., Inc. trademark. • **Apple, an Apple Computer, Inc. trademark.

FCC Lightens Up

In an eleventh hour action, the Federal Communications Commission granted Tandy Corp. a reprieve by waiving technical standards and certification requirements for Model I interface devices. This last minute stay of execution covers only 0K, 16K and 32K expansion interfaces (#26: 1140, 1141, 1142) and allows Tandy to continue marketing and selling these RFI-prone units until December 31, 1981.

The waiver is in effect—providing several stipulations are adhered to by the manufacturer. The net increase in interference potential of the TRS-80, Model I must not be greater than 6 dB when a noncomplying interface device is attached. A 6 dB increase is a net gain of 2X. Also, Tandy is allowed to manufacture up to, but not more than, 30,000 interface units during the waiver period. Finally, the Commission also stipulates that interfaces manufactured under the waiver may not be attached to any personal computer manufactured after January 1, 1981.

This last stipulation carries some interesting implications. It is possible that those attempting to purchase an expansion interface in the coming year will be required to show proof of Model I purchase predating January 1, 1981.

Commission motivation in granting Tandy's waiver stems from the desire of all parties involved to allow Tandy to live up to the commitments it has made to customers with regard to the availability of expansion interfaces for the Model I system. In addition, the FCC's action is an effort to ease the strain placed on all personal computer manufacturers by the stringent new radio frequency interference specifications taking effect January 1, 1981.

In the same session in which Tandy won its reprieve, both Apple Computer Inc. and the Heath Co. received extensions of their personal computer certification periods from January 1, 1981 to April 1, 1981. In each case, these manufacturers pleaded extreme financial hardship in efforts to gain certification. Claiming that they would have to close their plants if the FCC persisted in its insistance that they meet certification requirements by January 1, 1981, both manufacturers succeeded in getting the commission to back-off. The onus of adding to the already serious unemployment problems in America was apparently too much for the commission to bear. ■

by Chris Brown
80 Staff

NEW PRODUCTS

Edited by Chris Crocker

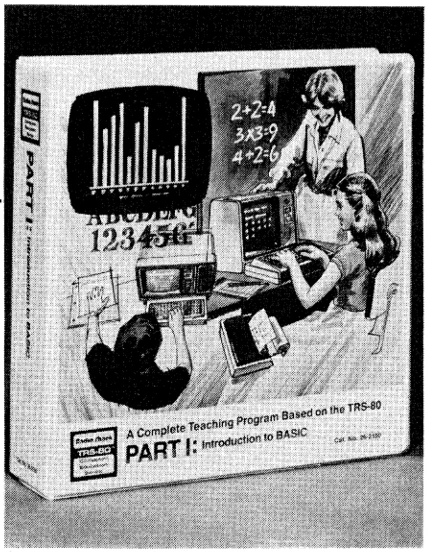
Radio Shack BASIC Learning Package

Introduction to BASIC Programming, Part I is a part of the Computer Education Series, a classroom package from Radio Shack designed for students in their first experience with computer programming.

The package includes a teacher's manual, a set of transparencies for overhead projectors and 25 student workbooks. The program included requires one or more 4K or 16K Model I Level I or II TRS-80s. The package is available at Radio Shack dealers for $159.95.

Also available from Radio Shack is *The Science Fair Story of Electronics*, a comic book available free to teachers, youth groups and students. For more information, contact Tandy/Radio Shack, 1800 One Tandy Ctr., Ft. Worth, TX 76102.

Reader Service ✓165



Radio Shack's Introduction to BASIC Programming, Part I

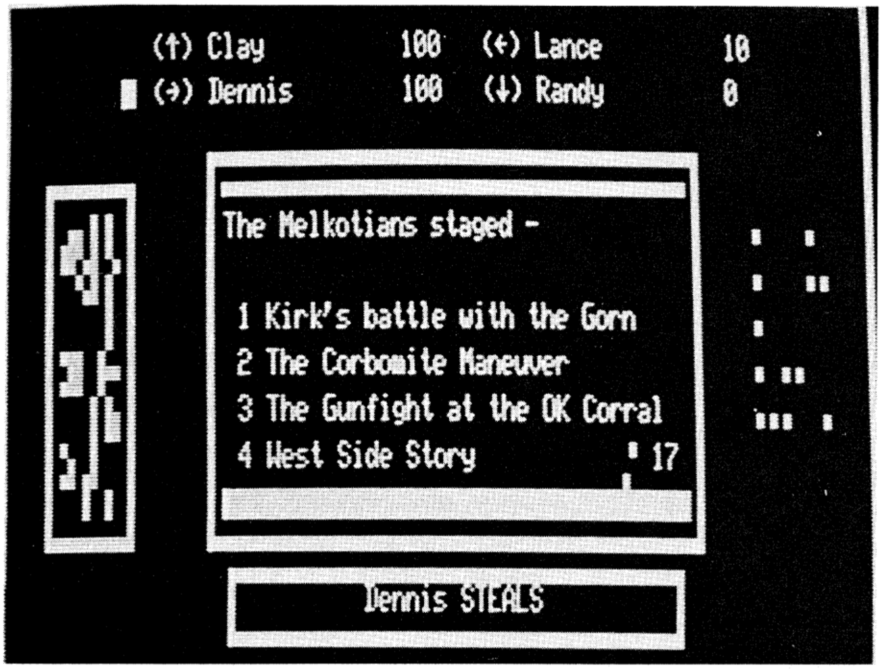
Radio Shack's Science Fair Story of Electronics

Game Tests Knowledge

The Wizard is a question and answer game authored by Richard Taylor. The game quizzes up to four players in four pre-programmed categories using the TRS-80 Model I system. A built-in utility program allows users to design their own data base.

No prices were released. For more information contact Programs Unlimited, Jericho Products, Inc., 125 South Service Rd., Jericho, NY 11753.

Reader Service ✓334



The Wizard

Education Newsletter

Microcomputers in Education is a monthly newsletter carrying reviews of educational software, new product announcements, reviews of books and magazine articles and industry news. A yearly subscription costs $15. *Queue's* Catalog #3 is also available, listing educational software from over 40 suppliers for the Apple, Atari and TRS-80. The catalog costs $8.95.

For more information, contact Queue, 5 Chapel Hill Dr., Fairfield, CT 06432.

Reader Service ✓161

Math Education Programs

Two educational programs are designed to develop mathematical reasoning and mathematical concepts. The Esti-

The New Products section is intended to inform our readers of new products on the market. All information in the section is taken from product releases sent by manufacturers. Because of the volume of product releases, we cannot attest to the quality of the products listed.

mation Game develops number sense and estimation in whole number computation. The Distance Game provides experience with two and three-dimensional graphing.

The programs are designed for use in grades three through nine. Each is on cassette for $9.95. For more information, contact Educational Programs, P.O. Box 2345, W. Lafayette, IN 47906.

Reader Service ✓340

Software, Hardware Catalog

Simutek's TRS-80 Users Catalog lists software from Simutek, as well as other manufacturers. The catalog also lists hardware, and is available from Simutek, P.O. Box 13687, Tucson, AZ 85732.

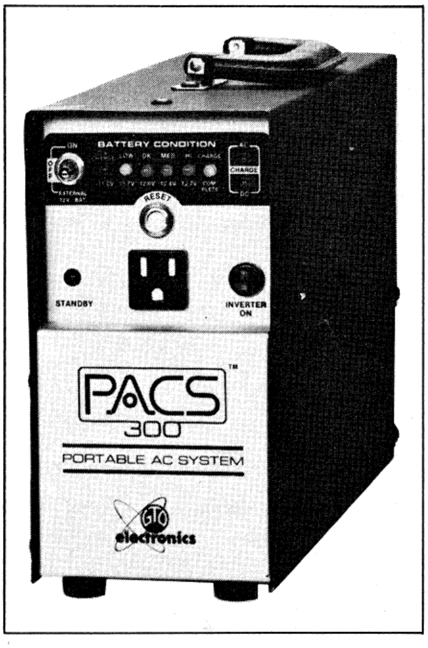
Reader Service ✓173

System Provides Portable Power

PACS-300A is a portable power system which provides 110 volt ac power in the event of a shutdown, and PACS-300B provides portable ac power at remote locations. PACS provide 150 Watts continuous or 300 Watts surge power.

PACS are available from GTO Electronics, 430 Ritt Street, St. Peter, MN 56082.

Reader Service ✓183



PACS Portable Power System

Demagnetizer Is Self-contained

The Maxell HE-44, a tape head demagnetizing cassette, is powered by a single micro button cell. The HE-44 achieves complete demagnetization in one second, according to Maxell.

The HE-44 is available for $24 from Maxell Corp. of America, 60 Oxford Dr., Moonachie, NJ 07074.

Reader Service ✓171

Realty Program Calculates Mortgage

Custom Tailored Software's mortgage pre-qualification system figures an applicant's limits for a mortgage through either VA or FHA methods of financing a home. The operator gives the program the total income and any outstanding debts, along with savings. The system figures in income taxes, the current interest rate, and closing costs. It then gives an optimum mortgage amount for that customer.

The package costs $199. For further information, contact Custom Tailored Software, Inc., 93 Old Homestead Rd., Wayne, NJ 17470.

Reader Service ✓180

Newsletter Gives Buying Advice

A monthly newsletter for small businesses is designed for users with little prior background in computers. The newsletter covers topics in the micro and mini field and provides advice on buying computers.

Subscriptions cost $35 from Computertalk Associates, Whitpain Office Campus, 1750 Walton Rd., Blue Bell, PA 19422.

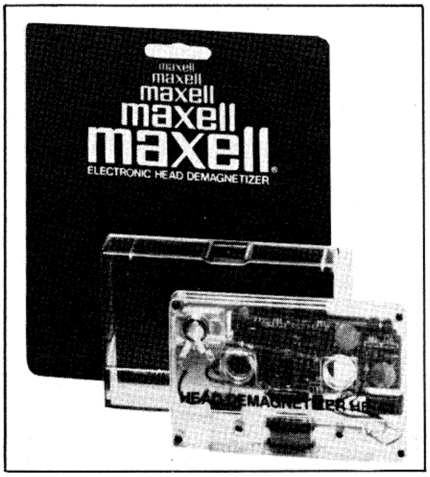
Reader Service ✓181

Catalog, Data Base Manager

The Micro Yellow Pages list software from Micro Architect, Inc. for TRS-80 Models I, II and III. The catalog also lists software for CP/M and Heath systems.

Also available from Micro Architect is IDM-V, an interactive data management package including a data base manipulation program, a report writer and report generator. IDM-V requires TRSDOS, two disk drives and 48K. The package costs $149. Catalogs are available free from Micro Architect, Inc., 96 Dothan St., Arlington, MA 02174.

Reader Service ✓182



Maxell Head Demagnetizing Cassette

Hot Line Answers Questions

A free Hot Line service gives technical aid to owners of Charles Mann & Assoc. software. The service is capable of voice and digital communications and is open 60 hours per week. The Hot Line can be reached at (714) 365-8558.

Also from Charles Mann & Assoc. are three catalogs listing applications software, including office management programs, accounting and financial management and professional applications. For more information, contact Charles Mann & Assoc., Micro Software Division, 7594 San Remo Trail, Yucca Valley, CA 92284.

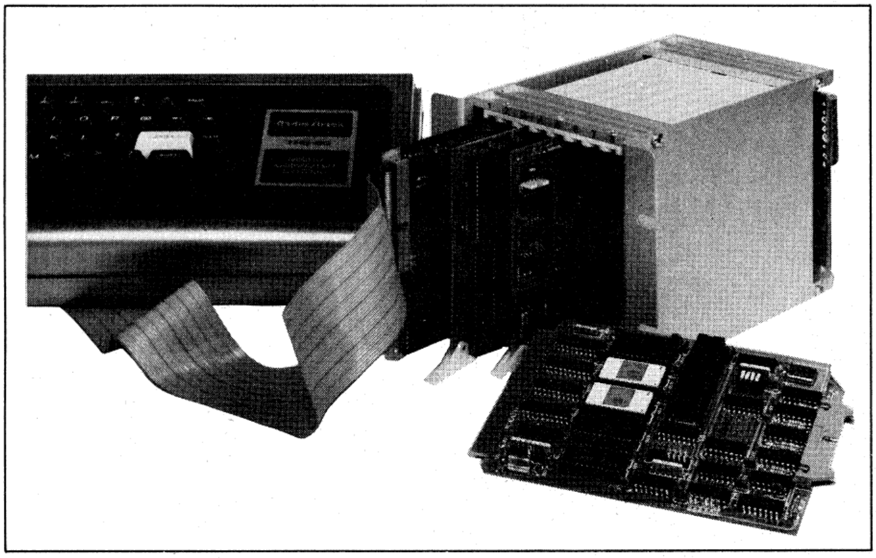
Reader Service ✓184

Model I and II Data Management

The Data Organizer is a database management system for the TRS-80 Models I and II. Programs include create, sort, select, print, edit and three label printing.

The system requires 48K or 68K and at least one disk drive. The Data Organizer is on disk for the Model I at $200 or for the Model II at $250. The manual alone may be purchased for $5 from Comprehensive Microcomputer Systems, Inc., 3132 N. Broadway, Chicago, IL 60657.

Reader Service ✓160



Xitex STD Bus I/O Cards

Standard Bus I/O Cards

A set of STD Bus I/O cards from Xitex Corp. emulate the keyboard, cassette and video functions of the TRS-80. When used with existing Z-80 CPU and RAM cards, these cards will operate TRS-80 compatible software.

These STD Bus I/O cards are distributed by QC Microsystems, P.O. Box 401326, Garland, TX 75040.

Reader Service ✓328

Program Aids Accountants

Datawrite is a client write-up system for accountants that operates on CP/M compatible microcomputers. The system incorporates several journal options and report writer capability and allows an accountant to format client statements. Datawrite is designed for either floppy or hard disk systems.

For more information, contact Dataword, Inc., 1404 140th Place N.E., Bellevue, WA 98007.

Reader Service ✓331

Directory Alphabetizer

DOS Alphabetic Directory is a machine language program that alphabetizes directories. Directories are listed in four columns of 12 rows, allowing for 48 file names.

According to Terra 80 Software, the program works with any DOS and any drive number. DOS Alphabetic Directory costs $14.95 from Terra 80 Software, 4660 Willens Ave., Woodland Hills, CA 91364.

Reader Service ✓336

Software Tutors in Chemistry

Formulas & Equations is a set of software for high school and college freshman general chemistry courses. The three-program set provides tutorial instruction and problems in chemical reaction equation balancing, stoichemistry (mass relationship) calculations and in determining the validity and structure of organic compounds.

The programs are on one cassette and require a 16K Model I Level II TRS-80. The package costs $25 from Custom Comp, P.O. Box 125, Branson, MO 65616.

Reader Service ✓185

Model I Source Text Editor

Editor is a full screen assembler source text editor for the TRS-80 Model I. Editor allows creation of new source files or editing of files previously created by Radio Shack's tape or disk EDTASM.

No prices were released. Editor is available from Computer Applications Unlimited, P.O. Box 214, Dept. 14E, Rye, NY 10580.

Reader Service ✓337

Pocket Computer Newsletter

The Pocket Computer Newsletter reports on news and product reviews concerning pocket computers such as the TRS-80 Pocket Computer. The newsletter is published ten times annually.

The ten-issue subscription price is $20 in the U.S., $24 in Canada, $30 elsewhere. A sample issue costs $2. For more information, contact the Pocket Computer Newsletter, P.O. Box 232, Seymour CT 06483.

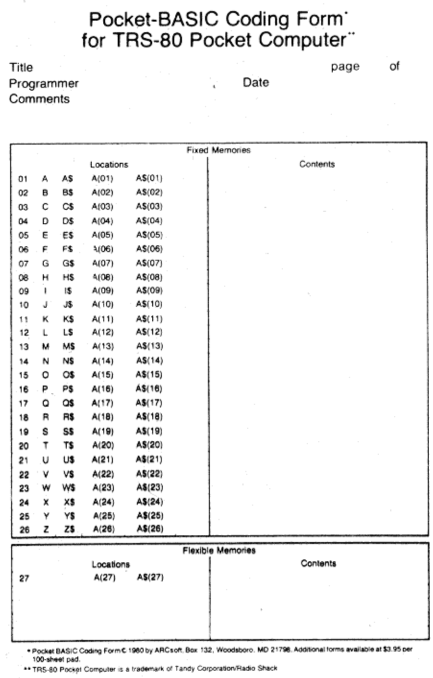
Reader Service ✓168

Pocket BASIC Programming Aid

The Pocket BASIC Coding Form displays the TRS-80 Pocket Computer's fixed memories side-by-side with space for listing their contents. It also has room for the programmer to label and list flexible memories. The reverse of the form is ruled for 30 horizontal program lines, each divided by 80 vertical columns for identification of available spaces in the standard TRS-80 Pocket Computer input memory.

Pads are available from Arcsoft Publishers, P.O. Box 132E, Woodsboro, MD 21798. Fifty sheets cost $3.95. Pads of 100 sheets are $4.95.

Reader Service ✓162



Pocket-BASIC Coding Form*
for TRS-80 Pocket Computer**

Title page of
Programmer Date
Comments

Fixed Memories

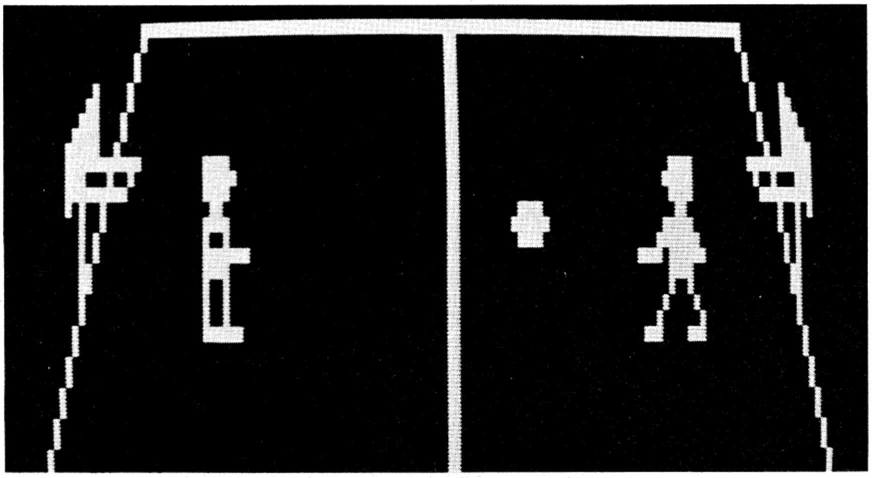
Locations					Contents
01	A	A$	A(01)	A$(01)	
02	B	B$	A(02)	A$(02)	
03	C	C$	A(03)	A$(03)	
04	D	D$	A(04)	A$(04)	
05	E	E$	A(05)	A$(05)	
06	F	F$	A(06)	A$(06)	
07	G	G$	A(07)	A$(07)	
08	H	H$	A(08)	A$(08)	
09	I	I$	A(09)	A$(09)	
10	J	J$	A(10)	A$(10)	
11	K	K$	A(11)	A$(11)	
12	L	L$	A(12)	A$(12)	
13	M	M$	A(13)	A$(13)	
14	N	N$	A(14)	A$(14)	
15	O	O$	A(15)	A$(15)	
16	P	P$	A(16)	A$(16)	
17	Q	Q$	A(17)	A$(17)	
18	R	R$	A(18)	A$(18)	
19	S	S$	A(19)	A$(19)	
20	T	T$	A(20)	A$(20)	
21	U	U$	A(21)	A$(21)	
22	V	V$	A(22)	A$(22)	
23	W	W$	A(23)	A$(23)	
24	X	X$	A(24)	A$(24)	
25	Y	Y$	A(25)	A$(25)	
26	Z	Z$	A(26)	A$(26)	

Flexible Memories

Locations			Contents
27	A(27)	A$(27)	

* Pocket BASIC Coding Form © 1980 by ARCsoft, Box 132, Woodsboro, MD 21798. Additional forms available at $3.95 per 100-sheet pad.
** TRS-80 Pocket Computer is a trademark of Tandy Corporation/Radio Shack

Arcsoft Pocket BASIC Coding Form



Acorn Software's Basketball

One-on-One Basketball

Basketball is a one-on-one game program for the TRS-80 Model I Level II by John Allen. The program allows one or two players; the single player option offers five levels of difficulty.

The program costs $14.95 on cassette or $20.95 on disk from Acorn Software Products, Inc., 634 N. Carolina Ave S.E., Washington, DC 20003.

Reader Service ✓327

Catalog Lists Engineering Software

A catalog from Microcomp lists software for engineers and surveyors. Programs are listed in such areas as hydraulics, hydrology, surveying, structural design, and business applications.

The catalog is available free from Microcomp, P.O. Box 965, Solana Beach, CA 92075.

Reader Service ✓177

Model I Interface Adaptor

The Model 488-80B enables a TRS-80 Model I with 16K RAM and Level II BASIC to be used as a GPIB-488 controller. A machine level driver program provided with the Model 488-80B on tape or disk, interacts with Level II, Level III and Disk BASIC.

The price is $225. For more information, contact Scientific Engineering Laboratories, 11 Neil Dr., Old Bethpage, NY 11804.

Reader Service ✓333

Utility Scans Disks

Trackcess is a utility for TRS-80 owners with 48K disk systems. Trackcess reads and writes sectors, and will also read or write entire tracks, whether on standard TRSDOS disks, irregularly designed disks, or protected disks. The utility scans and analyzes the disks, as well.

Trackcess is available for $24.95 from The Alternate Source, 1806 Ada St., Lansing, MI 48910.

Reader Service ✓170

Program Calculates Balanced Cattle Feed

A ration balancing plan calculates least-cost rationing for beef cattle. The program, which is written in machine language, will include up to 100 feed ingredients in calculation.

The program operates on a 48K Model I TRS-80. A demonstration tape for 16K is available for $5. The program and manual cost $650 from Agricultural Software Consultants, Inc., 1706 Santa Fe, Kingsville, TX 78363.

Reader Service ✓169

Catalog Disk Files

FLOPYCAT/BAS is a Model I program for building, maintaining and listing catalog files of disk collections. The program is designed to run on TRSDOS 2.3 and reads the directory of any compatible disk. Also available is a companion utility program, DISKNAME/BAS, which allows changing the name or date of TRSDOS compatible disks.

FLOPYCAT/BAS is available on formatted disk (or cassette, on special order) for $30. DISKNAME/BAS, when purchased with FLOPYCAT costs $10; otherwise it is priced at $15. Both programs are available from Marvin W. Plunkett, Microcomputer Systems Consultant, 1641 Northwest Rutter Lane, Roseburg, OR 97470.

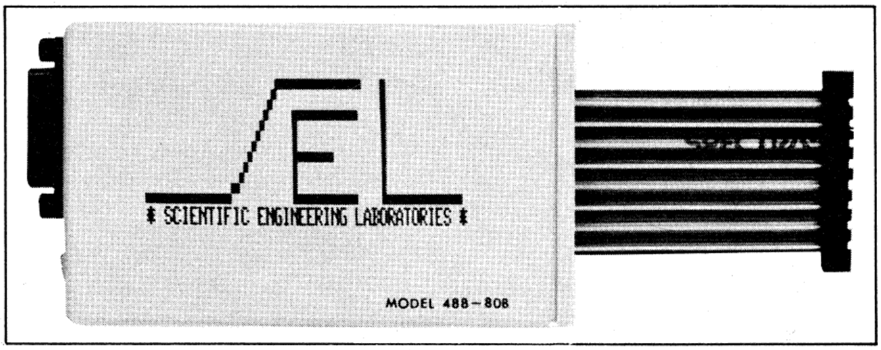
Reader Service ✓332

Inventory Control, Accounts Receivable

The Inventory Control System will handle up to 5000 items with full integration to Taranto & Associates' Invoicing and General Ledger Systems. Also from Taranto is the Balance Forward Accounts Receivable System, capable of handling up to 2000 customers and 11,000 monthly transactions. The system keeps transactions separate and includes a mailing and shipping label program.

Both packages are for the TRS-80 Model II and cost $399 each from Taranto & Assoc., P.O. Box 6216, 121 B Paul Dr., San Rafael, CA 94903.

Reader Service ✓326



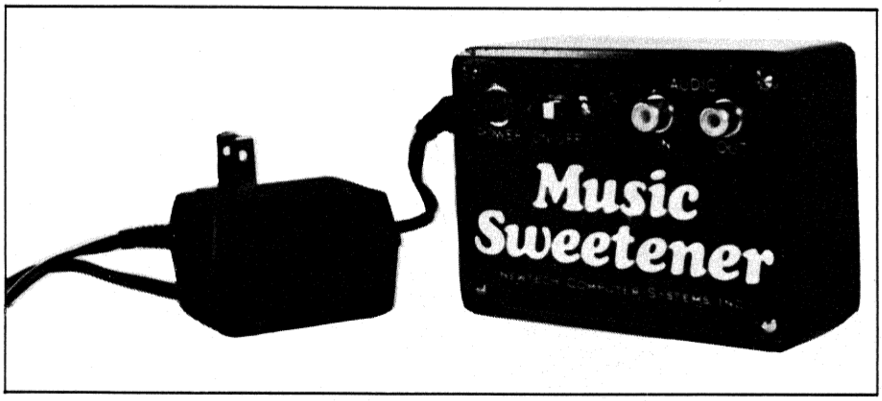
The Model 488-80B

Filter Improves Music Quality

The Music Sweetener is a low-pass filter designed to improve the sound quality of filterless commercial and homemade digital-to-analog-converter music synthesizers. The Music Sweetener attenuates the high frequency sampling noise and reduces distortion.

The filter is inserted between the music peripheral and audio amplifier. The package includes instructions and an ac adapter. The Music Sweetener costs $41.95 from Newtech Computer Systems, Inc. 230 Clinton St., Brooklyn, NY 11201.

Reader Service ✓166



Newtech's Music Sweetener

Business Software Runs on CP/M

Business Software from Univair operates on CP/M systems for the Model II. Programs available include Insurance Agency, Medical Management, Dental Management, Legal Time Accounting and Real Estate Multi-List.

For further information, contact Univair, Inc., 10327 Lambert International Airport, St. Louis, MO 63145.

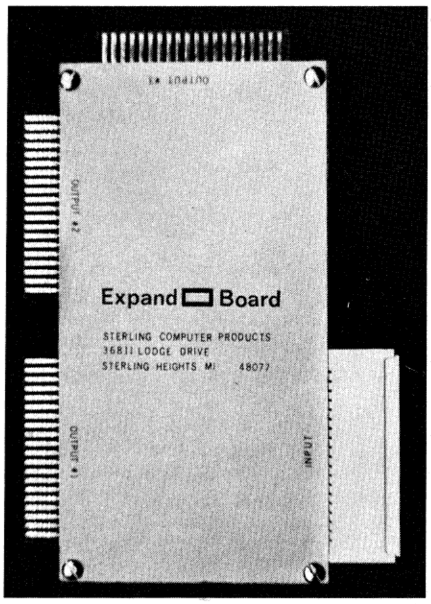
Reader Service ✓179

80 EPROM Information

A 14-page booklet called the *80 EPROMMER* provides product descriptions of the 2708, 2516, 2716 and 2732 EPROM programmer for the TRS-80. The booklet also serves as a do-it-yourself guide for construction.

The booklet comes with schematics, a parts list, and software listings from Graves Manufacture and Service, P.O. Box 306, Lake Bluff, IL 60044.

Reader Service ✓174



Expand-O-Board

Screen Dump And Graphics Control

CRT-Dump dumps the entire screen contents onto the programmer's cassette, disk, printer or high RAM. CRT-Art is a graphics controller for developing screen art work.

CRT-Dump is available on cassette for $9.95 or disk for $14.95. CRT-Art is available on cassette for $14.95 or on disk for $19.95. Both are available from JWMB Programming Corp., 507 E. 21st St., Lumberton, NC 28358.

Reader Service ✓330

Three-port Extender

The Expand-O-Board is a three-port extender for the TRS-80 Model I, allowing the connection of additional peripherals to the keyboard unit or expansion interface.

The Expand-O-Board costs $29.95 and is available from Sterling Computer Products, 36811 Lodge Dr., Sterling Heights, MI 48077.

Reader Service ✓329

Program Plays Cribbage

Cribbage Master is a cribbage game program featuring graphics display of the player's cards. The program pegs its own points in play, and entries are made with a single keystroke.

The program operates on a TRS-80 Level II with 16K and costs $12.95. For more information, contact Manhattan Software, P.O. Box 35, Pacific Palisades, CA 90272.

Reader Service ✓325

BASIC Cross-Reference Utility

Reference/Mod-II is a BASIC cross-reference utility for the TRS-80 Model II. The program allows display or printing of sorted cross-references to numbers or variables within a program and references to BASIC keywords.

The program exists as part of the BASIC system in a separate area of memory. Reference/Mod-II is available for $50 from Racet Computes, 702 Palmdale, Orange, CA 92665.

Reader Service ✓176

High Capacity Mini Floppy Drive

Apparat has teamed dual sided 80 track mini floppy drives with modification patches to NEWDOS/80, making each disk appear as a single volume, but with 405K of storage. The drive plugs directly into the expansion interface with no modification, according to Apparat.

The first drive, with case, power supply, interface and documentation (including NEWDOS/80 patches) is available for $839 from Apparat, Inc., 4401 Tamarac Parkway, Denver, CO 80237.

Reader Service ✓175

Let your fingers do the teaching.

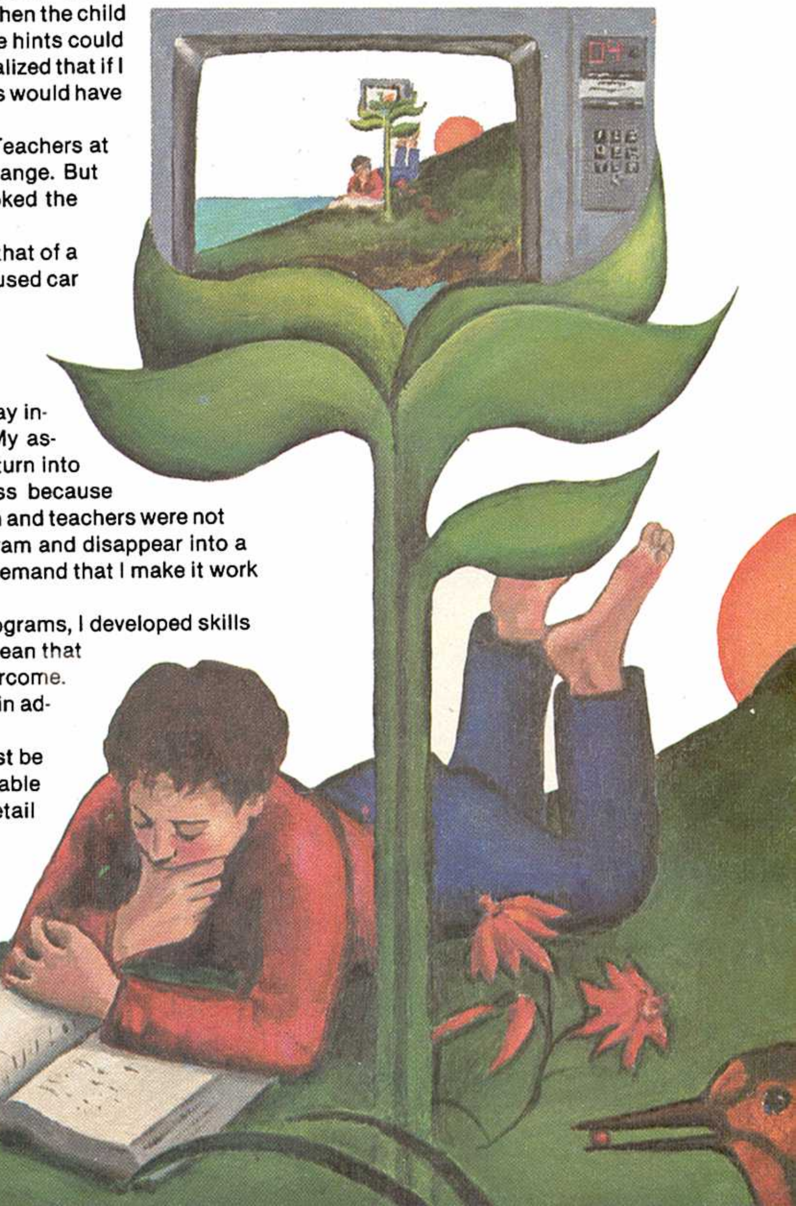
Programming for Education

J. I. Weintraub, 690 Mtn. View Rd., El Cajon, CA 92021

When I saw the little five year old respond to the TRS-80 as if it were the ultimate toy, I decided that educational programs for elementary school children could be a fantastic success. My optimism disappeared when the child suddenly stopped responding. Gentle persuasion and not so subtle hints could not convince her to answer the simple questions on the screen. I realized that if I wanted first graders to use and enjoy my programs, those programs would have to meet the interests and technical levels of the children.

After that early disappointment, I turned to the professionals. Teachers at the local school are enthusiastic; they work hard and accept change. But someone must have filled the media center with tear gas or hooked the new TRS-80 up to a live wire, because no one came near it!

My role with both the teachers and the students changed from that of a doting father, offering his children an expensive toy, to that of the used car salesman, trying his best techniques to sell the worst car on the lot.



A Different Experience

It's quite different writing programs for people you see every day instead of strangers who respond to ads in *80 Microcomputing.* My assignment was to produce programs that five year olds could not turn into spaghetti because of their innocence, or that would be useless because teachers were afraid the computer would bite them. These children and teachers were not unknown faces in distant locations; I could not sell them a program and disappear into a mailing list. If the program quit, they knew where to find me and demand that I make it work as I promised it would.

As a result of this front line exposure to writing educational programs, I developed skills and procedures that eventually proved successful—by which, I mean that the shyness of the child and the reluctance of the staff were overcome. Simply stated, writing educational programs requires many skills in addition to the art of programming a computer.

The following checklist summarizes the specific areas that must be addressed to produce educational programs that will be acceptable for use in elementary schools. These points will be discussed in detail over the next three months.

- The program should contain complete documentation.
- Input traps must be provided to avoid sending the program off on the wrong track, or hanging it up indefinitely.
- The program must be flexible to allow for differences in student needs.
- You must provide a method of coping with the child who does not respond to an input statement.

"My optimism disappeared when the child suddenly stopped responding."

Documentation

Did you ever load one of your own programs that you had not seen for months, and discover you had forgotten exactly how it was supposed to work? If this has happened to you as often as it has to me, you understand the need for explicit documentation. I use the word documentation to mean all the instructions and background information essential to using the program. This doesn't include technical information regarding the programming or the equipment.

You might offer a printed manual with the program, or you can include the documentation within the program itself. Printed manuals are usually more complete, but they can get filed away, mislaid or lost.

The original user can get so familar with a program that he does not miss the manual, but later users will have to figure it out for themselves.

For that reason the information vital to the control of the program should always be part of the program itself.

Many programmers provide thorough documentation in REM statements. While this is sufficient for the average programmer, it is of little help to the layman who is not familiar with programming techniques. The non-programmer needs to see the information and instructions in order to act on them. The program should also allow users to skip any information with which they are familiar.

Since your average educator or student will know very little about the program itself, your instructions should be detailed and comprehensive. Try to make the screen display easy to read. Full 64-character lines are difficult to read—they will be neater and easier to read if you keep your lines approximately half the screen width. Double-space when ideas change, and use CLS (clear screen) frequently to display new concepts. Avoid printing near the top of the screen and try to center your material.

Sadly, experience proves that instructions on the screen are difficult to remember, probably because so many people do not really read what they see! You should provide a means for the user to review instructions if he or she so desires.

Write clearly and simply. Too many times I have read articles in computer journals that claim to be addressed to the beginner. But if these are examples of simple articles, I would have to place myself in the lowest of reading groups. After the introductory paragraphs I usually cannot understand what is being said. Read your material carefully, or have a novice read it, to be sure that *anyone* can understand it.

Input Error Traps

Surely you have answered a question on the screen that goes something like this: DO YOU NEED INSTRUCTION? You do, so you type YES, and the computer blithely ignores your request and goes on as if you had answered NO. Later you find out that you should have answered Y instead of YES. Or perhaps a prompt is provided: DO YOU NEED INSTRUCTIONS? (Y/N), and you accidentally strike the wrong key. These are only a few of the problems you can run into when the programmer neglects to provide input error traps.

When you are dealing with young children, input error traps become even more important. Not only are children prone to make errors, but they should not be required to worry about the mechanics of the keyboard or the techniques of the programmer. Any time the user must make a choice you should provide an input error trap. For example:

```
10 PRINT "INSTRUCTIONS:"
20 PRINT "TYPE 1 FOR THIS."
30 PRINT "TYPE 2 FOR THAT."
40 PRINT "TYPE 3 FOR THE OTHER."
50 INPUT A
60 ON A GOTO 100, 200, 300
100 PRINT "HERE YOU ARE AT THIS!":END
200 PRINT "HERE YOU ARE AT THAT!":END
300 PRINT "HERE YOU ARE AT THE OTHER!"
400 END
```

Any integer input other than 1, 2 or 3 bypasses line 60 and goes to line 100. A simple trap at line 65 would be:

```
65 PRINT "TYPE 1, 2, OR 3 PLEASE!":GOTO 10
```

A similar trap should be used for string (i.e., alphabetical) inputs:

```
10 INPUT "DO YOU WANT INSTRUCTION? (Y/N)";A$
20 IF A$ = "Y" THEN 100
30 IF A$ = "N" THEN 200
40 PRINT "PLEASE ANSWER Y FOR YES OR N FOR NO.":GOTO 10
```

Programs are often written:

```
10 INPUT "DO YOU WANT INSTRUCTIONS? (Y/N)";A$
20 IF A$ = "Y" THEN 100
30 GOTO 200
```

This program assumes that if the answer is not Y it must be no. This assumption saves memory space and is a little easier to program, but should the child type YES instead of Y, he will end up with a no-no!

I recommend using a prompt (Y/N) with every question in the program in which there are options. The prompt should also be associated with the answer, if at all possible. I can see no reason to substitute 1 for YES, if the computer is capable of reading strings.

Later I will deal extensively with the use of INKEY$ to facilitate certain kinds of input. While INKEY$ is a powerful tool, it creates the possibility of several other input errors. Simply stated, INKEY$ avoids the use of the enter key. It accepts any input and treats it as if it were part of the response. For example, if you made an error while typing the response and hit the left arrow to backspace, the computer would treat the left arrow as if it were part of the response. It would add its ASCII value to the ASCII values of the other letters or numerals and, obviously, produce an unwanted input. To add insult to injury, it would not backspace!

To avoid these problems, you must alert the computer by placing a trap in the routine. The ASCII code for the left arrow is 8; the code for the ENTER key is 13. You trap these by entering the following statements:

```
50 IF A$ = CHR$(8) THEN (see note)
60 IF A$ = CHR$(13) THEN (see note)
```

Finally, you should let the user review his input to decide if they are really what he intends. This is especially important when:

- Several input statements are required before further execution of the program. Many people do their thinking after they see the input on the screen, and then it is too late unless you provide a means of reentering the information.
- An erroneous input at that point will result in a computational error that may never be detected.

The following example is taken from an addition program in which the user is asked to enter the limits on the sizes of the addends:

```
10 INPUT "MINIMUM SIZE OF FIRST ADDEND";A1
20 INPUT "MAXIMUM SIZE OF FIRST ADDEND";A2
30 INPUT "MINIMUM SIZE OF SECOND ADDEND";B1
40 INPUT "MAXIMUM SIZE OF SECOND ADDEND";B2
```

"Did you ever load one of your own programs that you had not seen for months, and discover you had forgotten exactly how it was supposed to work?"

```
50 INPUT "DO YOU WANT TO CHANGE ANYTHING (Y/N)";A$
60 IF A$ = "Y" THEN 10 (All four inputs will have to be re-entered.)
70 IF A$ = "N" THEN 100 (Continue execution.)
80 PRINT "TYPE Y FOR YES OR N FOR NO.":GOTO 50
```

Provide Maximum Flexibility

Programs can be written with such a narrow field of application that their usefulness is severely limited. The result is restricted applicability and/or a need for several programs where one would do the job.

In a school setting, providing a multitude of limited applications programs is a monetary issue related to more than the purchase of programs. It actually relates to personnel salaries!

When different students use the computer one after another, different programs will have to be loaded. This can be time consuming, and the salary of the teacher loading the programs becomes a factor in the cost of computer operations. This can result in budget problems. The computer can be a wise investment because it requires a one-time expenditure that is soon amortized. School personnel cannot be required to monitor a computer during any period that a student is using it. If such a condition were necessary, I would suggest getting rid of the computer and letting that teacher tutor the child.

Programs should be flexible enough to minimize loading new programs when a different student comes to the computer. There are four ways to provide this flexibility.

- Teacher inputs, where the teacher defines the parameters of the program before the student uses it.
- Levels, where the student or the teacher simply selects the subroutine within a given program that is appropriate for the student.
- Timing loops, where the teacher determines the length of time the student will be given to respond.
- Self-cycling program, where the lesson automatically resets itself for the next student, avoiding the need for someone to be present after each student finishes.

Teacher Inputs

There are 100 basic addition equations from 0+0 to 9+9. There are three ways to limit the scope of the drill. For the advanced student, you will want a drill on all 100. Ask, DO YOU WANT ALL BASIC FACTS (Y/N)? If yes, provide randomly generated problems.

A second alternative is to drill on one family of equations, such as 9 + 0, 9 + 1, 9 + 2, etc. Ask, DO YOU WANT A SINGLE NUMBER FAMILY (Y/N)? If yes, ask WHICH FAMILY (1-9)? This input will set one addend to the level requested. The other addend is generated at random.

In the third case, the teacher sets the highest and lowest limits of the possible sums. Ask, MAXIMUM SUM OF . . .? and MINIMUM SUM OF . . .? After your random number statements, you insert two added statements, IF A + B > MAX THEN . . . and IF A + B < MIN THEN . . ., in each case returning to the random number generators.

If you provide a spelling program, it will, out of necessity, provide a limited number of words for the student to study. Once these words have been mastered, the program is of little use to the teacher or the student. Your program will be of much greater value if you include instructions for changing the words. Be specific. Tell the user the statement numbers that contain the data. Explain how to retype the statement number; type the words (so many to a line), with commas between them. Explain that all the old lines must be changed so that none of those words will be retained. Have the user check to be sure the program contains the same number of words it did before, and, finally, provide instructions telling how to save the program if the user decides he may wish to use it several times.

"Educational programs must be written in a manner that avoids monitoring."

Program Listing 1 illustrates a portion of an actual program in which first graders practice the order of the letters of the alphabet. The portion illustrated asks the child to respond with the letter that comes before the given letter. The teacher decides which letters the children will work on by responding to the input requests within the program.

Line 65 ensures that the inputs are legal. Line 125 determines the ASCII code for the desired response. Line 130 determines the letter following the required letter by adding 1 to the ASCII code and converting it back into a string value. Lines 140 and 145 use a time limit and the INKEY$ function, both of which will be explained in detail in Part II of this article.

Levels

One alternative to teacher inputs provides levels in programs in which the material gets increasingly more difficult or complicated. This lets brighter children move ahead faster, working at more difficult levels, while the slower learner works at the easier level until he has mastered it. One way of achieving this is to provide five levels of a drill along with a review. For example:

```
TYPE THE NUMBER OF THE LEVEL YOU DESIRE:
1. SUMS OF 0 TO 5
2. SUMS OF 5 TO 8
3. SUMS OF 8 TO 10
4. SUMS OF 10 TO 13
5. SUMS OF 13 TO 18
6. RANDOM BASIC FACTS
```

There are five ways to implement such a program:

First, the teacher may select the level and all students at that level work with the program that day.

Second, the program itself can keep a record of the student's errors. The problems that are missed are displayed or printed after the student finishes. The student copies them for later study and review. Based on this information, the teacher places the student at the appropriate level at his next sitting.

A third way records the student's progress on a data tape. You can keep a record of each response, or only a statement of the student's current level. The tape is reviewed at the end of the day, and a record is kept of each student's placement.

The fourth method evaluates the student's performance and tells him which level to work on next. The student maintains his own record card and refers to it each time he comes to the computer.

The fifth method uses a disk operating system to maintain records and to place the student at the appropriate level automatically.

My students are currently using a 200 level math program I developed that covers math skills from grades one to six. The child is placed at the level maintained for him by the computer. After the child has identified himself, the computer serves up the appropriate level. One child completed 28 levels at a single sitting!

I'M A BELIEVER !!

VTOS 4.Ø
by VIRTUAL TECHNOLOGY

SCOTT ADAMS SEAL OF APPROVAL

1) Large (8") Drive Support.
2) Double sided Drive Support plus 35, 40 & 80 track drive support
3) 80 Track drive support. (NOTE: all above drives may be mixed on any one system and can be configured at Sysgen time during or any Backup.
4) Double density drive support.
5) Winchester Technology fixed drive support.
6) Supports any combination of the above drives up to a max. of 8 drives.
7) Supports double-speed processor clock modifications. (Archbold for example)
8) FASTER! Improved overlay structure using ISAM accessing techniques, improves loading time by up to 1400%.
9) General purpose output spoolers of a true, symbiont design provide simultaneous output and program execution without any user intervention.
10) Keyboard Type-Ahead feature permits you to enter keystrokes before your program needs them.
11) User definable keys, all 26 letters
12) Built in Graphic String Packer lets you enter graphic symbols into a BASIC program from the keyboard through the use of the (Clear) key. The (Clear) key is simply held down (Just like the (Shift) keys) during other keystrokes and voila GRAPHICS.
13) Dated files. - All files are accompanied by the date of their last modification. (creation or write)
14) Marked files. - All files are accompanied by a 'mark' is they have been modified since they were last backed up. This permits the BACKUP utility to copy only those files which have actually been updated since a previous backup.
15) File transfer by class. Allows transferring of all files of a similar directory classification such as /CMD, /BAS, /PCL, etc.
16) Built in SYSTEM command contains lower case display driver, screen print, break key disable, blink-cursor, disk drive stepping rate and motor-on delay modifications, and more.
17) Users may SYSGEN a custom VTOS system configuration containing special I/O drivers, device LINKing and ROUTEing, SPOOLing and DEBUG tasks, etc. which will be automatically loaded during the BOOT process without requiring a more lengthy AUTO and CHAIN procedure.
18) Non-BREAKable AUTO and CHAIN commands.
19) Wild-Card DIRectory. Permits you to locate all files of a certain classification such as '/BAS'. Uniformly file size in K (1024 bytes) regardless of drive type. "DIR D" would give you all of your files that start with D.
20) Dynamic file name defaults in APPEND, COPY, and RENAME commands allow you to specify only minimal information about file names.
21) COPY and APPEND commands execute up to 300% faster.
22) ALLOCate command for pre-allocation and non-releasibility of file space. File space will never shrink if this option is used.
23) MEMORY command for directly setting upper memory limit.
24) Variable length file support is incorporated which automatically blocks short user data records both within a sector and across sector boundries thereby taking maximum advantage of disk file space.
25) No security disk needed to make backups or to run the system.
26) Though manyO/S bear his design and code, VTOS 4.0 is the ONLY FULLY APPROVED OPERATING SYSTEM by Randy Cook! And it's FANTASTIC!
27) Endorsed by SCOTT ADAMS & LANCE MICKLUS.

"I Love it !!. . .It's really an incredible O/S. It' just great! Now I see why people who have seen it say they are now believers. I know I am."

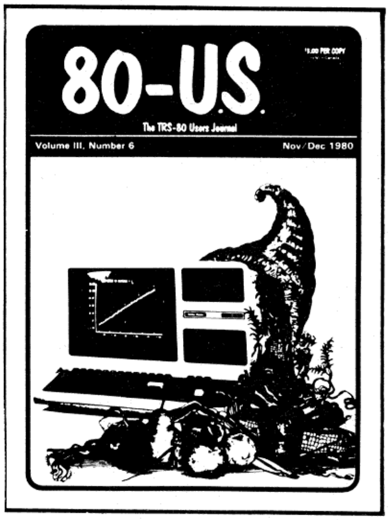
LANCE MICKLUS

VTOS 4.Ø

VTOS 4.Ø
Operating System Diskette with Operator's Guide
$99.95

VTOS 4.Ø
Master Reference Manual
$29.95

VTOS 4.Ø
Combination - 4.0 disk, Operator's Guide, and Master Reference Manual
$125.00



80-US - NOVEMBER/DECEMBER 1980
". . . without a doubt, the most flexible system around."

80-US - NOVEMBER/DECEMBER 1980
"Cataloging all of the 'can's' with this system is a near impossibility. It is so flexible, that its limits have hardly been touched."

80-US - NOVEMBER/DECEMBER 1980
". . . I didn't feel that the DOSPLUS (a competiting operating system) really had any speed advantage."

AVAILABLE FROM THE FOLLOWING DISTRIBUTORS OR FROM YOUR LOCAL COMPUTER STORE

DEALER INQUIRIES INVITED.

5% Discount Just For Mentioning This Ad. (Valid month of this publication ONLY)

QS QUALITY SOFTWARE DISTRIBUTORS
11234 Park Central Pl Suite C
Dallas Texas 75230
(214) 692-1055
Micronet - 70130,203
SOURCE - TCC293

Adventure INTERNATIONAL ✓213
ADVENTURE INTERNATIONAL
Box 3435, Longwood, Fla. 32750
(305) 862-6917 - Voice
after 8:00 - same number
as FORUM 80. (SOURCE - TCC957)

SMALL BUSINESS SYSTEMS GROUP
Main St. and Lowell Road
Dunstable, Mass. 01827
(617) 692-3800 - Voice
(617) 692-3973 - FORUM 80
Micronet - 70310,236

VTOS and VTOS 4.0 are registered trademarks of VIRTUAL TECHNOLOGY, INC. - Dallas, Texas 75234

"Sadly, experience proves that instructions on the screen are difficult to remember, probably because people do not really read what they see!"

Program Listing 1

```
5 'LISTING NO. 1
1Ø CLEAR5ØØ
15 CLS
2Ø PRINT"LISTING 1"
25 PRINT STRING$(63,61)
3Ø PRINT"EXAMPLE OF TEACHER INPUT.."
35 PRINT STRING$(63,61)
4Ø PRINT"ENTER THE 8 LETTERS"
45 PRINT"YOU WANT THE CHILDREN TO ENTER."
5Ø PRINT"PRESS <ENTER> AFTER EACH ONE."
55 FOR A=1 TO 8
6Ø PRINT A;":";:INPUTH$(A)
65 IFASC(H$(A))<66 OR ASC(H$(A))> 9Ø THEN 8Ø
7Ø NEXT A
75 GOTO 85
76 E=12
8Ø PRINT"ILLEGAL INPUT.":GOTO 6Ø
85 CLS
9Ø INPUT"HOW MANY SECONDS FOR EACH RESPONSE (SUGGEST 10)";T
95 IF T<3 OR T>15 PRINT"KEEP IT BETWEEN 3 AND 15.":GOTO 9Ø
1ØØ CLS
1Ø5 PRINT"THIS PORTION IS FOR THE STUDENT.":PRINT"PRESS ENTER.":IN
    PUT A$
11Ø IF B$=CHR$(13) THEN 165
115 T1=T*125:G=G+1:F$=INKEY$
12Ø IF G>8 THEN 175
125 M=ASC(H$(G))
13Ø M$=CHR$(M+1)
135 PRINT" ?      ";M$
14Ø FOR X=1 TO T1
145 A$=INKEY$: IF A$=""THEN NEXT X
15Ø IF X>T1 PRINT"TIME'S UP!":GOTO 17Ø
155 PRINT" ";A$;"      ";M$
16Ø IF A$=H$(G) THEN 165 ELSE 17Ø
165 PRINT"CORRECT.":FOR X= 1 TO 1ØØØ: NEXT: GOTO 11Ø
17Ø PRINT"TRY AGAIN!":FOR X=1 TO 1ØØØ: NEXT: G=G-1:GOTO 11Ø
175 PRINT@7*64,"THAT'S ALL FOR TODAY!"
18Ø PRINT"GOOD-BYE FOR NOW."
185 PRINT"GO BACK TO YOUR CLASS."
19Ø FOR X = 1 TO 2ØØØ: NEXT   'USE 9ØØØ IN ACTUAL PRACTICE
195 G=Ø: GOTO 11Ø
```


Another of my programs deals with skills involved in using reference books such as the dictionary, encyclopedia, and card catalog. It is set up in 36 levels.

Timing Loops

Students cannot be relied on to respond promptly. They often will not guess if they are not sure of the answer. Also, unless you provide the response, I DON'T KNOW THE ANSWER, what are they to do when they just don't know? Sometimes they get confused and forget what to do next. By providing a time limit for their response and programming the computer to react to a non-response, most of these problems can be overcome.

There are two variables to consider when using time limits; first, the variation between the more able and the slower students, and, second, decreasing the response time as the students move from the learning level to the mastery level.

The time limit is programmed by using a variable in a loop. See Program Listing 1, lines 90, 115, 140, 145, and 150. Line 90 allows user input. Line 115 converts seconds to increments for a do-loop as explained on page 54 of the TRS-80 User's Manual for Level I. Line 140 sets the do-loop, using the variable T1. Line 145 waits for a response T1 times, and, if none is forthcoming, line 150 branches the program to an alternative.

The alternative can be TRY AGAIN, as on line 170, or it can be a hint. In a program where there is text material for the student to read before he responds, you can return to the text and ask the student to study it again.

The timing loop can be used in a reading program to control the rate at which the words appear on the screen.

```
10 DIMA$(100) (Make it large enough to include the number
   of words in the story.)
20 ONERRORGOTO 110
30 INPUT"HOW MANY WORDS PER MIN";M
40 M = 400*60/M
50 A = 1
60 READA$(A)
70 FOR J = 1TOM:NEXT:PRINTA$(A);"";
80 A = A + 1
90 GOTO 60
110 PRINT:PRINT:PRINT"END OF STORY."
120 DATA NOW, IS, THE, TIME, FOR, ALL, GOOD, MEN, TO
200 END
```

Use ON ERROR GOTO in line 20, to save you the effort of counting the exact number of words in the story. When that number is

"One alternative to teacher input provides levels in programs in which the material gets increasingly more difficult or complicated."

reached, the ON ERROR routine diverts the program to line 100.

You may find that line 40 does not produce the exact rate you asked for. Individual computer units appear to operate at a slightly different rate. I suggest you enter 100 words and use a stop watch to check the rate. Increase or decrease accordingly.

Self-Cycling:

Educational programs must be written in a manner that avoids monitoring. Self-cycling programs meet this requirement once the teacher variables have been entered. As each student completes his turn, the program automatically returns to the beginning of the student section of the program.

The student program has to end with proper cues. If you say, "Good-bye, John. Time for the next student. Press ENTER." John will press ENTER. The program will restart before the next student arrives. A more effective approach is to get John away from the computer while it is in a long do-loop:

```
GOOD-BYE, JOHN.
TIME TO GO BACK TO YOUR ROOM.
```

I now introduced a long do-loop, like FOR X = 1 TO 9000:NEXT, followed by a GOTO the beginning of the student program.

The student program should begin with a greeting and an instruction to press ENTER, thereby initiating the program.

```
10 CLS
20 PRINT @ 6 * 64, "HI THERE!"
30 PRINT "PRESS <ENTER> WHEN YOU ARE READY."
40 GOTO (beginning of lesson)
```

(Are you using the 'PRINT AT' technique illustrated in line 20? By using 64 as a factor, you select the line on which you want to print by setting the line number as the other factor. Also if you want to indent a certain number of spaces you add that number to the product: 6 * 64 + 10.)

The teacher also needs to know where the student program begins. I insert a line at the end of the user portion that says, "The next instruction is for the student."

A second point which needs explanation when automatically recycling a program is the strategic placement of dimensioning statements. Page 4/4 of the Level II manual covers this potential problem, but inadequately.

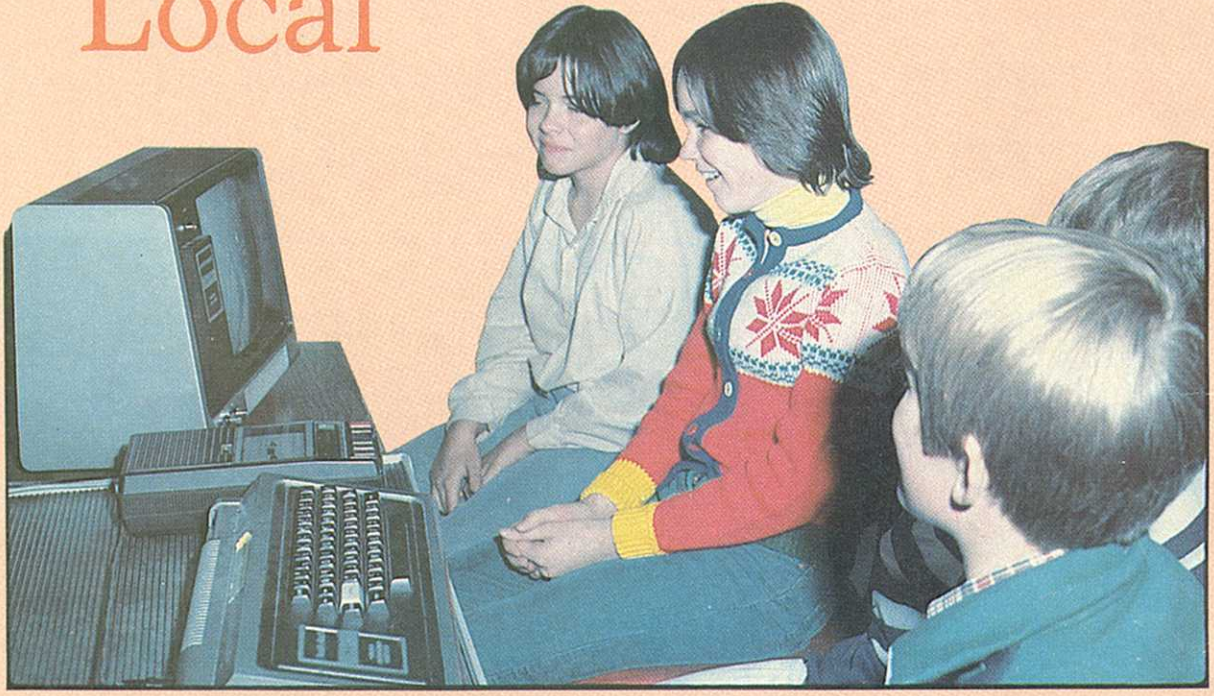
The DIM statement must be placed outside the student program unless you use a CLEAR statement as well. However, the CLEAR statement will reset all variables to zero. If you have a record keeping system, CLEAR will reset the variable in it to zero before the next student starts. The alternative to CLEAR is to insert C = 0:N = 0 at the appropriate spot in the program.

If you use data statements, you must insert a RESTORE statement somewhere between the end of one student's turn and the beginning of the next. RESTORE simply allows the computer to reuse the same data. See page 3/10 of the Level II manual for details.

Next month: INKEY$ and audio.■

✓ Reader Service—see page 242

Project Local



Students work at a TRS-80 located in one of the Westwood grammar schools.

A classroom computer project with a 13 year history.

by Pamela Petrakos
***80* Staff**

Computer assisted learning was the objective of educators from five Massachusetts towns 13 years ago, when they started their innovative program known as Project Local.

"Laboratory Program for Computer Assisted Learning," Project Local, is somewhat unique among educational institutions. Certainly, it was one of the first in this country.

In 1967, the towns of Westwood, Natick, Needham, Wellesley and Lexington in a joint effort, requested federal funding under (the now defunct) Title III of the Elementary and Secondary Education Act (ESEA) in order to investigate the use of computers in learning, especially in the area of mathematics.

"Project Local was designed," said Jim Pender, a math teacher and one of the authors of the proposal, "to determine whether math could be taught more effectively with the aid of a computer, especially in algebra."

Dr. John Tobin, Chairman of the Board of Directors for Local since its inception said the project was initially begun to study how computers might facilitate learning in mathematics; however, it was also started with the idea that computers were an ideal learning tool to help improve problem solving among students.

Telecomp Pilot

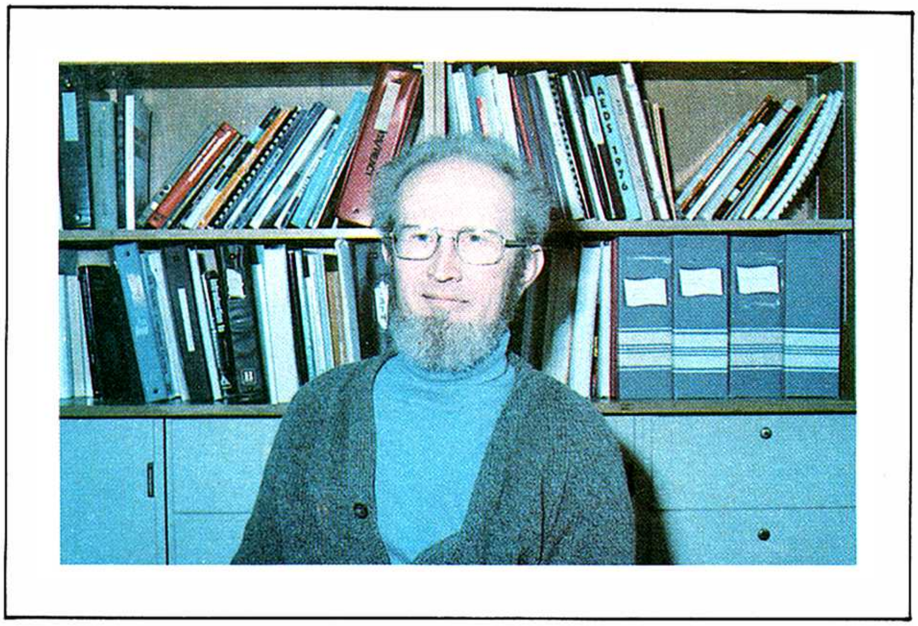
Robert Haven, Director of Local from 1967 until this year, said that the project grew out of a pilot in which Westwood and Lexington were involved. These two had obtained a federal grant under Title IV of ESEA. The initial grant enabled them to lease teletypes which were hooked up to a computer in Cambridge (Bolt, Beranek and Newman, Inc.) as part of a service called Telecomp.

In its first year of operation, Local leased the same time-sharing service (Telecomp). However, the second year, after a grant receipt of $98,000 the five-school systems (known as Sponsors of Project Local) obtained five minicomputers from Digital Equipment Corp. (PDP-8Is) on a lease-purchase agreement. One mini was placed in each of the five school systems.

Over the following three years the minis were paid off. The initial $98,000 was not spent exclusively on the five computers, said Haven. Half of the money went towards the purchase agreement and half went towards setting up the Project Local office and hiring a staff.

The goal of its first three years, said Haven, was primarily to evaluate materials and techniques that were used in the preceding Westwood-Lexington project.

After the first three years, federal funding ended, but the project was considered suc-



Robert Haven has been Director of Project Local since 1967. He resigned this year after relocating to Maine.

cessful enough that enthused educators decided to support the program themselves. Their objective, said Haven, "was to provide services to school systems who wanted to use computers to teach."

In 1970, Local became a not-for-profit corporation and received its financial support (as it does to this day) entirely from school systems to which it provides services.

Concurrently, the five original sponsoring towns decided to open up membership in Project Local to other school systems.

In the next few years (1970-74) the number of sponsoring members grew to a total of ten towns. According to Haven, most of these additional members had their own hardware and others began time-sharing by hooking up teletypes, either their own or leased equipment, to the two PDP-8Es that Local had acquired.

However, in 1974 it became obvious that computers could not be maintained without a resident hardware expert. (Sponsoring members, as a part of their contract with Local, are entitled to hardware maintenance service.)

Pamela Elsworth, Educational Computer Consultant and Inservice Instructor for Project Local said, "Every time one broke down, we (Bob Haven and myself) were running to one of the schools—not to repair them—but to decide which vendor to call."

Project Local opened up a Central Computing Facility in 1974, making time-sharing available to all its members. Of the ten members at that time, roughly half utilized the time-sharing option for which the two PDP-8Es were used. Eventually a third PDP-8E was acquired. (These three Digitals were later sold. Currently one is serving the Westwood School System where it is interfaced to eight teletypes in a math lab.)

Its First Micro

In 1977, Project Local acquired its first microcomputer, a PET. Six months later they bought a TRS-80. "The PET," said Haven, "was the first free-standing computer Local had. It was strictly for the use of Local and was placed in the demonstration center, to show schools what could be done with them."

In 1979, school systems (both members of Local and neighboring towns) began to acquire microcomputers, said Haven. Schools financed these micros in a number of ways—their own budgets, Title IVB (Media Funds), Title VI (Vocational Funds) and Parent-Teacher Associations.

Sponsoring schools continue to contract with Project Local for inservice training, maintenance and supplies.

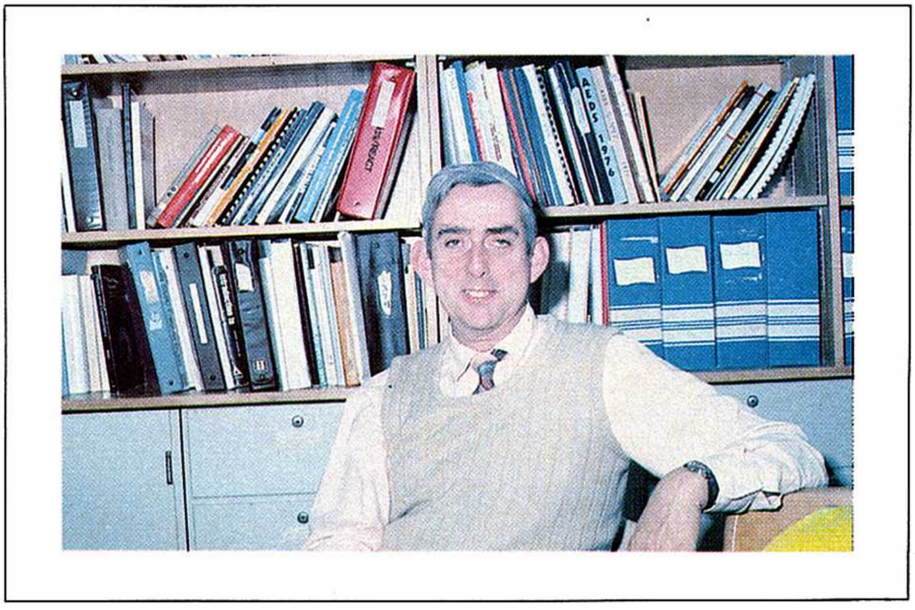
When a school system becomes a sponsor of Local it takes on part of the financial responsibility of running it. Annual dues are determined by a formula. The Board of Directors, which is composed of Superintendents and/or Assistant Superintendents from sponsoring schools, arrives at an expense budget for the coming year. Each sponsor decides what they will need for that year, ranging from instruction to supplies. The final budget is a total of what all sponsoring schools will need.

Each member school is allocated a certain portion of the total according to its size and the extent of its computer program.

Last year, Local started an associate member program, called Lamp. Based on the premise that a successful instructional computing program requires expertise and investment (which is often too costly for an individual school to afford), for a flat annual fee of $750, associate members can have a number of services available to expose their students to computer assisted learning.

The main differences between sponsoring and associate members is that first, and most important, sponsors are represented on the Board and thus, direct Project Local. Secondly, sponsors receive all inservice training free, whereas associates pay a reduced fee. Thirdly, sponsors are entitled to receive services at their schools, whereas associates must go to the Local office for services.

The services that Project Local offers to both sponsors and associate members are quite extensive:



Jim Pender is one of the original authors of the proposal for Project Local. He is a math teacher at Westwood H.S.

● A toll-free telephone consultation to members between 8 and 5, weekdays, to answer any question or resolve any problem concerning instructional computing.
● Cooperative microcomputer purchasing program. Once a year, all Local members have an opportunity to participate in a joint effort to acquire microcomputer equipment and supplies at low prices. Lists of estimated equipment needs are compiled and bid on. Winning bids are published in a catalog which is distributed to all member districts.
● A microcomputer demonstration center, located in Westwood High School has a variety of hardware (including TRS-80s, an Apple, a PET, a Compucolor, a KIM, a Digital DECwriter II, and a Teletypewriter) for teachers to experiment on. The demonstration center also has a library of over 2,000 educational software programs and literature. Educators from Local schools can visit by appointment during the week.
● Instructional software information services which include a software directory and an instructional software exchange which helps teachers make contact with colleagues in other districts to trade, buy or sell software.
● A cooperative evaluation of computer oriented instructional materials. Each of Local's member districts may designate up to three teachers, preferably in different subjects and levels, to act as evaluators of programs, textbooks and resource materials. Reviews of materials are consolidated and published by Local several times per year in a periodical called *Courseware Review.*
● Local also offers instruction for teachers whether they are novices or have some experience in computing. Elsworth conducts classes twice a year during winter and spring semester. Courses in Microcomputer Orientation, Introduction to Programming in BASIC, Introduction to the Use of the Computer in the Elementary Classroom, Intermediate Topics in BASIC Programming and Selecting Microcomputer Hardware and Software for Instruction are on the curriculum.
● The Project Local Newsletter *Local Link*, is published five times a year with information on computer education at Local headquarters, in Local schools and elsewhere in the U.S.

> *"Haven said that there was some resistance among teachers, because 'once you integrate computers into a math or science course you are talking about a change in teaching habit. . .' "*

Often, complaints are heard of the lack of resources specifically dedicated to educational computing. Below, are a few organizations and publications that we came across that are directly concerned with computers in the classroom.

CERC—Computer Education Resource Coalition, represents several organizations in the Boston area that provide services and information to teachers interested in using computers in the classroom.

CERC
c/o TERC
8 Eliot Street
Cambridge, MA 02138

BCS—Boston Computer Society, has a number of member users groups that meet regularly, with featured speakers on a variety of subjects.

BCS
17 Chestnut Street
Boston, MA 02107

TERC—Technical Education Research Center, established a Computer Resource Center to provide information and training for educational uses of microcomputers at the pre-college level. The center has a variety of different microcomputers and a sampling of educational applications (both software and literature) for educators to try.

TERC
8 Eliot Street
Cambridge, MA 02138

School Microware—A Directory of Microcomputer Software. Compiled by Robert Haven, past Director of Project Local, it has over 500 listings of programs and packages for use in the classroom. The cost is $20 which includes three updates during the year.

P.O. Box 246
Dresden, Maine 04342
(Also available through Project Local)

Queue—A Catalog of Educational Microcomputer Software provides descriptions of several hundred programs for the PET, Apple and TRS-80. Listings are grouped by computer, subject matter and grade level. The cost is $8.95.

Queue
5 Chapel Hill Drive
Fairfield, CT 06432

Project Local
200 Nahatan Street
Westwood, MA 02090

---

80 *also found several publishers in the educational field that might interest you.*

Classroom Computer News is a newsprint half-tabloid in its first year. It is directed to teachers and administrators using or thinking of using microcomputers in the classroom. It contains software reviews, applications and news stories.

Editor: Lloyd R. Prentice
Intentional Educations
80 Brighton Avenue
Allston, MA 02134

T.H.E. Journal—Technological Horizons in Education is a slick magazine published six times a year. Though not specifically dedicated to microcomputers, T.H.E. Journal carries a more eclectic mix of articles on the impact of technology in education and on society.

Publisher: Edward W. Warnshuis
Editor-in-chief: Dr. Sylvia Charp
Information Synergy, Inc.
7 Spruce Street
Acton, MA 01720

The Innovators

Project Local is only one of the revolutionary programs across the country experimenting with computer-assisted learning.

The Huntington Project, for example, also began in 1967 and was initially funded by the National Science Foundation (NSF). The objective was to help 26 Long Island school systems get started in instructional computing.

The Minnesota Educational Computing Consortium (MECC) was formed in 1973 as a joint project of all public education agencies in the state. The consortium promotes and supports educational computing projects throughout Minnesota.

All of these diverse programs share one common objective: to initiate and perpetuate computer literacy. In this objective, Project Local has been quite successful.

First of all, according to Haven, with the ready availability of microcomputers has come an influx of computer programs in schools. "However, whether this is translated into greater computer literacy depends on what the schools do in the way of familiarizing teachers with computer-assisted learning," said Haven.

Computers Help Learning

How do teachers react to computer-assisted learning? Most important of all—have computers helped students learn more effectively?

Haven said that there was some resistance among teachers, because "once you integrate computers into a math or science course you are talking about a change in teaching habits. There was some reticence as far as teachers using computers as a teaching tool. However, there is no question that there are enough teachers that are interested; this is obvious because all of the hardware in the demo center is being used."

Haven also said that the use of computers definitely does have an affect on student thought processes. "It helps them to take a more systematic approach to problems," said Haven. "One thing that has been shown over and over in studies is that, first of all, learning with the aid of a computer is more efficient, that is, more is learned in a shorter period of time. And secondly, it improves a student's attitude towards the subject area that the student is involved in."

Pender thinks that the way a computer helps a student learn is perhaps the most important aspect of the whole project. Pender said, "It makes a student logical, systematic, organized, insightful and creative, which normally can't be taught, (for example) in a math course. Computers also teach problem solving, analysis, and organization. In fact its better than a course in logic. There is no way a teacher can teach all of these aspects; they often get bogged down just trying to teach the basics."

What is in the future for Project Local? There is a fear among the staff and the board that over a decade of work may go down the drain. Tobin said because of the recently passed Proposition 2½ in Massachusetts, many towns can no longer afford to be members of Project Local. Currently, there are four sponsoring members, and there is a possibility, said Tobin, that all four may drop out by the end of next year. Local has appealed for funds from the state and from private institutions, such as the Ford Foundation and the National Science Foundation. They are sympathetic, said Tobin, but have yet to respond with the much needed money that Local needs to stay alive.

The future is cloudy, said Tobin. "It's a question now of whether we will just maintain with a minimum of staff and equipment or whether we can continue to move forward." ■

Two views of the experiment at Rosemount.

Classroom Computing: Genesis of a Program

Dr. Lee A. Droegemueller
School District 196
1445 Diamond Path
Rosemount, MN 55068

Are computers in elementary and secondary schools just another fad that will disappear as other educational gimmicks have in the past? Is computing only for mathematics teachers and their special students? These are questions that school personnel have had to face since the advent of the microcomputer in the classroom and a number of successful time-sharing programs in schools across the country.

Because of the widespread social impact of microprocessor technology the professional staff and the Board of Education in the Rosemount, MN school system have endorsed the use of computers in school. The position of the Rosemount district regarding computer instruction follows:

- Computer education is basic education, meant for all students.
- All teachers must be involved for maximum success.
- Computer materials and software must be easy to understand, and based on sound learning principles.
- Equipment must be easy to use, cost effective and not more complex than the task requires.
- Equipment must be expandable for future needs.
- Instruction should be interesting and exciting to ensure continued interest.

The success of the Rosemount program is based on heavy use of the computer as an instructional tool. The program requires teachers in the district to learn how to operate a computer and to understand the

Continued to p. 80

Norman Bell
4548 Keweenaw Drive
Okemos, MI 48864

The computer revolution has gone to school. Never before has so much computing power been available at such low costs. Consequently, many more teachers have access to computers, far more than even in the wildest of predictions ten years ago. The problem, then, is not how to get computers into the schools, but what to do with them once they become available.

Computer use in the classroom seems to develop in one of two ways. One situation may be described as follows: some teachers have gained access to computers, either through time-sharing or a school's acquisition of microcomputers. The teachers are interested in learning how to use computers and shortly find themselves surrounded by a group of very able students, sharing like interests. The group solidifies by becoming a computer club or some other such organization which offers a high degree of freedom. The teachers are now acting as sponsors. However, as time progresses, the group, because of their ability to specialize, become more and more sophisticated and fewer in number. Eventually the membership's growth rate becomes almost stagnant.

The second situation includes a very small number of teachers who for one reason or another have been assigned the task of familiarizing students with computers. These teachers are readily identifiable, since for the most part they are "running scared," as they have had all too little experience with computers and practically no background in the area. They just happened to be assigned by administrators. In this second situation, there are not usually many students involved since the teachers themselves want to figure the system out before trying to introduce it to the students.

Both of these situations have a high degree of similarity. First, the groups start small and often stay small. Second, the majority of teachers and students in the school are left ignorant or become "turned off" to learning about computers due to the lack of interest of the "experts" in teaching them. Often times, teachers also lack a plan and materials to instruct others about computers.

Consequently, any strategy for introducing computers in the classroom must take into account these all too typical situations.

The Rosemount Program

Several years ago, in Rosemount, Minnesota, these two situations were quite evident.

Though the concept of computer literacy was almost unheard of at that time, individuals at Rosemount identified it as the target and a plan was conceived.

The Boeing Computer Services had developed and was selling twenty half hour video tapes designed to provide computer literacy in management. These were fast moving, attractive presentations. But because Boeing's supplementary materials were inadequate for secondary schools, sets of teacher and student materials were developed. The materials include overhead transparencies for the teacher, student worksheets and a teacher manual to explain how the materials were to be used in the classroom. The design of the materials, made the teacher the key element in the entire process.

Research on the program's effectiveness

"The computer revolution has gone to school."

though it indicated some shortcomings, proved that the students and teachers did learn how to use computers. As a result of the program, Rosemount quadrupled its use of a local time-sharing service.

After one year of program operation, a major change in the computing world took place—low-cost microcomputers were introduced. Because of the increased availability of these computers at ever lower costs, Rosemount adapted its computer literacy program to focus mainly on the micro and its most common language, BASIC.

Instructional materials were developed for this task, and again the philosophy was to keep the teacher central to the presentation. However, one shift was made. Rosemount initiated the program at the elementary level among third, fourth and fifth graders. Again materials including overheads, student worksheets, and teacher guides were developed and made available to a group of volunteer teachers. Because of the nature of the materials and the completeness of the teacher guide, little inservice training appeared to be needed.

Half-hour Lessons

The lessons took about one-half to three-quarters of an hour to present, and then the same amount of time was allotted to students to take quizzes and experiment.

The first five lessons taught computing concepts using a four-function, ten-key calculator as "computer." Ten lessons then followed with direct focus on the Radio Shack TRS-80. These ten lessons cover the many concepts of the BASIC language—arithmetic, looping, graphics, subroutines and numeric arrays.

The program again received a favorable review. Student attitudes were measured. Tests showed that students gained not only a respect for computing, but also viewed themselves as able individuals, since they were capable of using such powerful equipment.

Later, both Rosemount and other locations used these same materials with learners ranging from lower elementary through post-doctoral. It appeared that when individuals have just about equal ignorance of a subject, the materials used to teach them may be similar, regardless of age.

In the lessons developed, five components were consistently maintained. First, in each lesson an overview, provided the learner with a general outline of the lesson. Following the overview was a set of objectives which provided the learner with a list of concepts and skills to be acquired. The third element of each lesson was a set of structured notes, designed to help the student focus on the material as it was presented by the teacher. The fourth component was an instructional quiz. Quiz results were immediately fed back.

The fifth and final component was a hands-on activity at the TRS-80. Included in these activities were running previously recorded programs, entering and running previously written programs, writing, entering, running, and debugging programs students had designed.

All these lessons have been thoroughly tested and revised, where necessary, and are now published by Radio Shack. The programs are entitled, *Part I, Introduction to BASIC, Part II, BASIC Programming, and Part III, Advanced BASIC Programming.*

Whether by design or chance, it appears that the teaching material provided enough assistance to teachers who had little or no computer knowledge so that they were able to instruct students as they themselves learned. So through this first phase of the Rosemount project, the situation of the small group of knowledgeable individuals becoming smaller had been reversed, and many students and teachers were gaining computer literacy.

With the problem of literacy at least partially solved, the next phase of the project was devoted to expanding the knowledge of teachers and students. The initial teaching model was used as the basis for additional sets of lessons on the BASIC language. These additional sets included advanced BASIC statements and commands, and an introduction to the disk operating system.

Current Phase

Mindful of its success, the current phase of the Rosemount project is focused on applications of computers to various subjects. For example, a series of lessons is being developed to instruct students in the use of computers in a business setting. The students are taught to use commercially available business programs. Similar lessons are being developed to teach general ledger, accounts payable, accounts receivable, payroll, inventory, and mail lists. This phase of the Rosemount project will prepare students for jobs with small companies which are managed with microcomputers.

In this entire set of programs, a key ingredient, and perhaps the most important factor of all, is that teachers have remained at the center of the entire process. They are able to learn, sometimes just ahead of the students, but always soon enough to be effective.

Consequently, the large numbers of teachers and increasingly larger numbers of students will be able to make effective use of the microcomputers of today and the more powerful systems that will be avaiable tomorrow.■

✓Reader Service—see page 242

"The popularity and efficiency of microcomputer networks has encouraged the district to investigate this new learning possibility."

impact that computers will have on the learning process.

Mechanical Aids

The track record of educators using mechanical aids to teaching has not been outstanding. The most used aid is probably the paper and pencil followed by the chalk board. Overhead projectors, filmstrips, 16 mm projectors, and video tape players fall somewhat behind in their use. Using record players in school, followed their use in the home by about five years. So it will be with computers. Most students, for example, had access to hand-held calculators long before they were endorsed by educators.

The Rosemount district has been aggressively committed to using technology to improve education since 1976. The need to educate over 1200 high school students in 1976 focused our attention on teacher-directed group instruction. We needed a structure to assure the student efficient use of his turn on the time-sharing terminal.

The inexpensive microcomputer seemed the best solution to our computer time-sharing problem.

The first classroom instruction program used were the Boeing Computer tapes, "Making It Count." The teacher introduced the material; structured work sheets were used and self-correcting quizzes were administered. This resulted in students learning the material, time-sharing usage went from less than 5000 minutes per month to over 50,000 minutes, and interest increased.

Prior to introducing computer instruction in the elementary school, the staff needed to identify the educational applications of the microcomputer. Under the direction of Dr. Norman Bell, a professor at Michigan State University, educational applications for the microcomputer were divided into three areas: it could be used as the object of instruction, the medium of instruction, or the manager of instruction.

In the elementary schools the computer was to be the object of instruction. Staff and students would focus on how to operate a computer, how the computer works and the impact computers have on society. In order to teach a computer's operation, students needed to understand the concept of input, control, processing, memory and output.

Ten-key Calculator

One of the best tools used to teach these concepts was the four-function ten-key calculator with memory. Five lessons were developed by Norm Bell for this purpose. The transition from the calculator to the 4K Level I TRS-80 was a natural one. Ten lessons were developed to be used in grades 3-5. (After two years of testing, this material was made available through Radio Shack Stores as the "Computer Education Series Part I, BASIC Programming.")

Similar programs are being used in the middle school. Radio Shacks Computer Education Series has now expanded to three parts and the computer as the object of instruction is being taught at all levels of the Rosemount school district. Once students and staff learned about the computer as the object of instruction, the stage was set for using the computer as the medium of instruction. At the high school level, computer laboratory courses grew, science simulations were investigated, word processing and business application programs were developed and computerized instrumental music instruction were used. At the middle school, one group of teachers have developed simulations for teaching math, English, and geography.

One of our more exciting experiments involves the use of Radio Shack's K-8 mathematics program. In two district schools using a network system, elementary teachers report students making significant gains using the program for drill. The automatic placement and recordkeeping systems are excellent. Five of the eight remaining schools have requested network systems for K-8 Mathematics.

Video Expansion

The potential of the microcomputer as a teaching aid is further enhanced when video tape and video disks are interfaced to it. To explore these possibilities, Rosemount had prototypes developed interfacing the TRS-80 with the Betamax Video Tape Recorder and the Pioneer Video Disk system. Instructional materials and teaching methodologies are now being developed.

The popularity and efficiency of microcomputer networks has encouraged the district to investigate this new learning possibility. The need for printed copies of programs written by students inspired the development of a prototype network that allows disk storage and printing from remote micros. This will be especially useful for teaching word processing reducing our need for disks, tapes and printers.

Another network being developed will allow remote micros, to access one or more video disk units. Students encountering problems in computer delivered instruction will be able to receive help or assistance in video format.

Using microcomputers successfully in the Rosemount school district was a result of the direction and planning that preceded our experimentation. Teachers were involved in development and testing, and the teacher remains central to the instructional process. Equipment was never purchased that was more complex than the task required. Administrative and consultant support was available for teachers when difficulty was encountered.

Microcomputers have provided new skills to be learned by students and staff; they have increased learning in pre-existing programs, and have created a new excitement in our schools. ■

These Next 4 Pages are for TRS-80* Owners ONLY!

The next 4 pages contain over 100 programs for your TRS-80. Whatever your interests, we have a software program for you. We list sections on Home/Personal, Business, Games, the Arts, Home Education, Utilities, Special Business, Flight Simulations, Electronics, Comp-U-Novels, and Popular Games. These programs can be purchased through your local Instant Software dealer, or you can call us directly using our toll free number. We ship our orders the same day we receive them. Browse through these 4 pages, we're sure you'll enjoy your selections. Remember: WE GUARANTEE IT!

UTILITIES

TRS-80 UTILITY I—Give your program that professional look. RENUM: Renumber any Level II program to make room for modification or to clean up the listing. DUPLIK: With this program you can duplicate any BASIC, assembly/machine language program, verify the data and record the program to tape. You can even record Level I programs on a Level II keyboard. **(T1) Order No. 0081R $9.95.**

TRS-80 UTILITY II—Change the drudgery of editing your programs into a quick, easy job. It includes: ● CFETCH: You'll be able to merge consecutively numbered BASIC programs into one program. It will also search through any Level II program tape and display the file names for all programs. ● CWRITE: Combine subroutines that work in different memory locations into one program. It works with BASIC and/or machine-language programs and will give you a general checksum to verify that your program hasn't dropped any bits. **(T1) Order No. 0076R $9.95.**

THE COMMUNICATOR—This package lets you transmit data over the telephone lines. The full ORIGINATE/ANSWER capability allows your TRS-80 to be controlled from a remote-based terminal, or allows two TRS-80s to "talk" to each other. You can transmit data or programs from home base to a remote terminal. There will be a simultaneous display of information on both video monitors. Requires a modem and RS-232 interface for each terminal. **(T1) Order No. 0126R $9.95.**

TERMINAL-80—Communicate with the rest of the world! These programs give you control of the RS-232 port of your Expansion Interface. You can connect one or more serial terminals to your TRS-80 and it will accept input from the RS-232 interface just as if it were entered from the keyboard. Your TRS-80 can also be transformed into a dumb terminal, for use in a time-sharing situation to talk with "big" computers via a modem. The LPRINT/LLIST commands will transfer a program to a receiving computer. Supports upper/lowercase, Level II & III control characters, and all functions such as CHR$. The baud rate is software controlled for your convenience. Requires an RS-232 interface. **(T1) Order No. 0130R $24.95.**

DISK SCOPE—Need to check out the contents of a disk? Then check out these three programs. ● FILELOC: If you know the name of the program or data file, FILELOC will show you which tracks and sectors contain that file, as well as how much memory the file takes when loaded into RAM. You can then print the information, search for a new file or exit to BASIC. ● CDISK: This utility and test program allows you to view any track and sector on your disks in ASCII, Hex and screen POKEs. It disregards all protection codes. ● PASSWORD: This machine-language program not only gives you a password for individual files, but for whole disks as well. **(T2) Order No. 0139R $19.95.**

DISK EDITOR—This machine-language program give you total access to ANY byte of information in ANY sector in ANY track of your disk! You can examine, alter, add and delete information with ease. You can even search for a specific string (up to 8 characters long). If you need hardcopy, use the LINEPRINT command to send a copy of the video display to your printer. It can be used with TRSDOS, NEWDOS and MicroDOS. Both the 35 and 40 track versions are included. **(T2) Order No. 0180RD $39.95.**

BPA (BASIC PROGRAMMING ASSISTANT)—BPA does three things for you: (1) It will list the variables used in a BASIC program. Optionally, it will list the line numbers where each variable appears; the variable-type symbol (string, integer, single or double precision); whether it is dimensioned and where it is changed. (2) It will produce a cross-referenced list of line numbers for GOTO's, GOSUB's and IF...THEN statements. (3) It will list the line numbers where a selected BASIC function word (e.g., INPUT, PRINT) is used. **(T1) Order No. 0203R $14.95.**

TLDIS & DLDIS—These two utilities are ideal for those who wish to decipher and/or modify machine-code programs. TLDIS (Tape-based Labeling DISassembler) and DLDIS (Disk-based Labeling DISassembler) are three-pass, label-assigning disassemblers that assign labels (where appropriate) to the routines in a machine-language program. Their output is almost identical to that of a hand-assembled source code. TLDIS can send the disassembly to cassette tape, DLDIS can send it to disk; both send it to the video monitor. Each version can be reassembled using Tandy's EDTASM or Apparat's disk extension of EDTASM, respectively. You can also send either disassembly to a printer (R/S parallel port). Because of the labels, it is a simple matter to change any object code program by disassembling it and making changes to the resulting source code, without losing track of the jump/load addresses. Labels start at "AA00" and increment up, in even numbered steps (AA02, AA04, etc.). The odd numbers (AA01, AA03, etc.) are left for your (optional) use in the reassembly. **TLDIS (T1) Order No. 0230R $14.95. DLDIS (T2) Order No. 0231RD $19.95.**

THE DISASSEMBLER—This is a single-pass, hex-notation that sends its output either to tape or to a lineprinter (R/S parallel port). The tape output is directly compatible with Tandy's EDTASM, so you can disassemble an object code tape and output it to tape, then use EDTASM to add, delete, change and re-assemble your new version. It displays the *displacement* and *absolute address* of any relative jumps made by the disassembled program. It also displays and ASCII characters used in an LD or CP opcode. It is relocatable and you can jump to memory locations and transfer control between Disassembler and other utility programs. **(T1) Order No. 0239R $9.95.**

There are over 300 Instant Software dealers throughout the U.S.A. and the world.

Go see your local Instant Software dealer before Christmas. He has a wide selection of Instant Software.

CODE—Minimum System Required

(T1) = TRS-80 Model I Level II, 16K RAM
(T2) = TRS-80 Model I Level II, 16K RAM with Expansion Interface 16 + K RAM and one disk drive
(T3) = TRS-80 Model II, 32K RAM

*TRS-80 is a trademark of Tandy Corporation.

SEE YOUR LOCAL INSTANT SOFTWARE DEALER OR **Just Call Toll-Free 1-800-258-5473**

Instant Software™

PETERBOROUGH, N.H. 03458

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

THE ARTS

COMPU-CAROLS—We are proud to present a selection of Christmas carols, played by your TRS-80. Just place an AM radio next to your keyboard and you'll be amazed at the quality of this computer-generated music. You'll hear AWAY IN A MANGER, NOEL, SILENT NIGHT, O LITTLE TOWN OF BETHLEHEM and eight more of your favorite carols. **(T1) Order No. 0036R $9.95.**

DOODLES AND DISPLAYS II—It includes: ● DOODLE PAD: Draw pictures and save them on cassette tapes. ● SYMMETRICS: An electronic kaleidoscope that's constantly changing. ● DRAWING: Like DOODLE PAD, but for the serious artist. Over 40 user commands. ● RANDOM PATTERN DISPLAY: The computer does the drawing, but those with itchy fingers can make alterations. ● MATHCURVES: Bring those geometry lessons to life. Six different geometrical curves on the screen of your TRS-80. ● RUGPATTERNS: Designs rug patterns with a choice of user or computer control. **(T1) Order No. 0042R $7.95.**

MUSIC MASTER—Includes these four audio treats: ● MICRO-ORGAN: This program changes your computer into a musical instrument, with a range of four octaves with three voices! You can play sharps and flats to imitate the sounds of an organ, harpsichord or piano. ● KALEIDOPY: Now you can have a computerized "player piano." Generate a symmetrical graphics pattern and then see it transformed into music. ● COMPOSER: Experiment with computer-generated music. You can select the length of the piece, its scale, and its tempo. ● KEYMANIA: Test your memory and your musical ear. One to four players try to repeat the melody that the computer creates. **(T1) Order No. 0084R $9.95.**

ELECTRONICS

HAM PACKAGE I—This versatile package lets you solve many of the problems commonly encountered in electronics design, including: ● BASIC ELECTRONICS WITH VOLTAGE DIVIDER: Solve problems involving Ohm's Law, voltage dividers and RC time constants; ● DIPOLE AND YAGI ANTENNAS: Design antennas easily, without tedious calculations. **(T1) Order No. 0007R $7.95.**

ELECTRONICS I—This package will not only calculate component values for you, it will also draw a schematic diagram. Included are: ● TUNED CIRCUITS AND COIL WINDING: Design tuned circuits without restoring to cumbersome tables and calculations; ● 555 TIMER CIRCUITS: Design astable or monostable timing circuits using this popular IC; ● LM-381 PREAMP DESIGN: Design IC preamps with this low-noise IC audio amp. **(T1) Order No. 0008R $7.95.**

QSL MANAGER—Ever looked at your log book and wondered if you sent a QSL card to the operator you worked last week? Maybe you sent a QSL but can't remembered getting one in return. The QSL MANAGER will help you set up a computerized log book that gives you *instant access* to your records. Make complete log entries which include: Date, Time, Call sign, Name, Band, both the sent and received Signal Reports, the Mode, whether a QSL card was sent or received and any remarks you want to add. The QSL MANAGER program has built-in editing features that let you keep your log book up to date. **(T2) Order No. 0151RD $19.95.**

HOME EDUCATION

MONEY MADNESS—You can experience the Raw Power of High Finance with two Big Money empires. ● MILLIONAIRES: Can you manipulate $1000 into a million dollars in fifteen years? It all depends on your strategy as you buy and sell properties, negotiate bank loans, collect rentals and accept sealed bids. ● TIMBER BARON: An in-depth experience of the timber business, from the time you cut the trees until your milled lumber reaches the market. These transactions are affected by those tough, unexpected eventualities that can upset the most careful plans. **(T1) Order No. 0156R $9.95.**

TEACHER'S AIDE—Now you can have the benefits of Computer Aided Instruction (CAI) in your own home. Create a question and answer lesson (up to 8000 characters), save the lesson on disk, then create an entire sequence of lessons. Perfect for parents, teachers and students who need the unlimited patience and undivided attention only a computer can provide. **(T2) Order No. 0214RD $34.95.**

GRADE BOOK—Teachers, now you can use the speed and accuracy of the computer to help calculate student grades. Just type in the grades for tests, quizzes, homework, classwork or special projects to calculate and display individual grade averages. You can also obtain a cumulative grade for a specific marking period—or a whole year! **(T1) Order No. 0050R $9.95.**

TEACHER—This program enables you to create your own tests, quizzes and exercises for the education of your children. You can even provide "graphic" reward for your children and provide hints for problem solving. **(T1) Order No. 0065R $9.95.**

LIFE—Create "living" organisms in which cells are constantly active. They are born, they multiply, they die. This computerized version of LIFE is based on the well known game popularized by Martin Gardner. You can create one-cell organisms, then observe their growth patterns. The library of commands give you unlimited versatility in the control of the cell patterns *you* have arranged. **(T1) Order No. 0078R $9.95.**

ARCHIMEDES' APPRENTICE—This two-part package will teach you the formulas used to find the volume of any solid object including paralellopipeds (cubes and rectangular solids), prisms, pyramids, cylinders, cones and spheres. It will show you on-screen diagrams of these figures, and present you with the formulas you'll need to compute their volumes. **(T1) Order No. 0092R $9.95.**

TYPING TEACHER—This complete seven-part package takes you from initial familiarization with the keys, through typing words and phrases, to complete mastery of the keyboard. Your computer can even become a bottomless page for typing practice. **(T1) Order No. 0099R $9.95.**

VIDEO SPEED READING TRAINER—Most people's reading speed is limited simply because they read individual letters or words. Now you can increase your reading speed and comprehension by reading whole words and phrases. This package will train your mind to quickly recognize numbers, words, letters and phrases. Start at any speed level at which you are comfortable and the computer will automatically advance you as your reading speed and comprehension increases. **(T1) Order No. 0100R $9.95.**

WORDWATCH—four different programs to entertain and educate. ● WORD RACE—race to the finish line of defining words correctly; ● HIDE N SPELL— find the misspelled word, then correct it; ● SPELLING TUTOR—a spelling lesson, but beware, the spelling may become unusual. There you have it, Wordplay × four = WORDWATCH. **(T1) Order No. 0111R $7.95.**

MIND WARP—This game includes: ● MIND TWIST: a Mastermind-type game with a twist. Try to guess the computer's secret digit sequence. ● MIND BENDER: A multi-level game where you must discover the computer's secret code. It's no mystery, the MIND WARP package is for puzzle lovers everywhere. **(TI) Order No. 0118R $9.95.**

INVESTOR'S PARADISE—Here are two programs to test your skill in the stock market. ● STOCK TREK: a stock market simulation in which you and up to five other investors buy and sell stocks. ● SPECULATION: a step beyond a mere simulation, you enter financial data on up to 25 real companies and start playing the market. This package lets you experience the thrills and triumphs of the stock market without risking a dime! **(T1) Order No. 0125R $9.95.**

IQ TEST—IQ TEST will administer and score an intelligence test in just 30 minutes. There are three equivalent tests, each consisting of 3 questions that survey your general knowledge and problem solving abilities. **(T1) Order No. 0157R $9.95.**

SPECIAL BUSINESS

BOWLING LEAGUE SECRETARY—This package is simple to operate and provides a dynamic reference to all the names of individual bowlers, their team numbers, scores, team names, league data and all necessary statistics. The system is highly adaptable, with 17 different scoring options that allow you to custom tailor the program to suit your league's special needs. And, if you even have any problems, simply type HELP and the program will give you an explanation of what information is needed—complete with a sample entry. The system puts at your fingertips all individual weekly scores, team cumulative scores, bowler cumulative scores and individual leaders in the following categories: high single, high series, high average and high points. **(T2) Order No. 0095RD $49.95.**

*TRS-80 is a trademark of Tandy Corporation.

SEE YOUR LOCAL
INSTANT SOFTWARE DEALER OR

Just Call Toll-Free
1-800-258-5473

BEGINNER'S RUSSIAN—In order to understand a foreign culture, you must know its language. The three programs in this package will give you on-screen displays of the characters of the Cyrillic alphabet, detailed instructions of their proper pronunciation and exercises that will have you recognizing and speaking simple Russian words. An excellent package for students, businessmen, scientists or anyone who is interested in learning the Russian language. **(T1) Order No. 0136R $9.95.**

EVERYDAY RUSSIAN—will acquaint you with the words for various foods, places to eat, signs and the names of stores—exactly what a traveller needs to know. Each of the three parts of the package not only teaches you the words but quizzes you on them as well. You can even practice typing in Russian. Discover the Russian language today! **(T1) Order No. 0137R $9.95.**

BOWLING LEAGUE STATISTICS SYSTEM—Keeps a computerized list of league data, team data and data for each bowler. Extremely flexible, it has a total of 16 different options to let you modify the program to suit your league's rules. It is easy to use and has a built-in "HELP" feature to aid you. **(T1) Order No. 0056R $24.95.**

HOME/PERSONAL

HOUSEHOLD ACCOUNTANT—Save with these two programs: ● BUDGET & EXPENSE ANALYSIS: It has nine sections for income and expenses and an option for quarterly/yearly reviews. ● LIFE INSURANCE COST COMPARISON: Compare the total costs of various insurance policies. Contrast term with whole life. It will store and display up to six prospective policies. **(T1) Order No. 0069 $7.95.**

PERSONAL BILL PAYING—You can keep a computerized list of ALL your bills (up to 22 accounts), each listed with its name, number, due date and amount owed. Individual accounts can be displayed with a month-by-month breakdown of payments (including check numbers) and current accounts can be seperated from inactive ones. It allows you to save the data to tape for future use. **(T1) Order No. 0103R $7.95**

NO MATTER WHAT YOUR NEEDS ARE, INSTANT SOFTWARE HAS A PROGRAM FOR YOU.

WRITE FOR OUR NEW INSTANT SOFTWARE CATALOG

We Guarantee It!

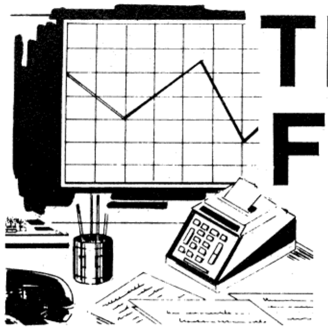
Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

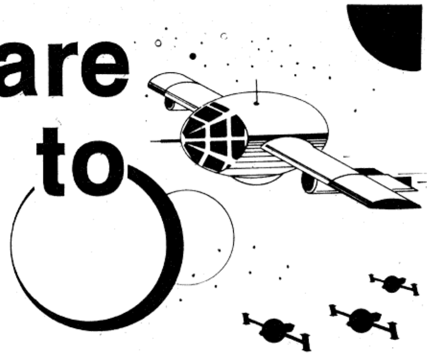
Instant Software™

✓2

PETERBOROUGH, N.H. 03458



TRS-80* Software From Function to Fantasy



POPULAR GAMES

BEGINNER'S BACKGAMMON/KENO—Why sit alone when you can play these fascinating games: ● BACKGAMMON: Play against the computer in a game that's sure to sharpen your skills; ● KENO: Enjoy this popular Las Vegas gambling game—guess the right numbers and win big! **(T1) Order No. 0004R $7.95.**

CHESSMATE-80—This versatile chess opponent gives you a choice of ten levels of play, from the "blitz" level (the computer has 3 seconds to move) to the infinity level (where the computer will consider every possible move—which could take years). This machine-language program is a conservative player and follows all the rules of international play. CHESSMATE-80 can teach you how to move and allow you to set up the board and play end games or special problems. CHESSMATE-80 battled Sargon II to a draw at two minutes a move and beat Microchess 1.5 in six moves. **(T1) Order No. 0057R $19.95.**

YOUR CRIBBAGE AND CHECKERS PARTNER—CRIBBAGE is a two-person game that you are sure to enjoy. This is NOT a tutorial—it is a game worthy adversary. CHECKERS: An old favorite which follows international rules, including multiple jumps. **(T1) Order No. 0068R $9.95.**

CARDS—A one-player package to let you play, with your computer, these famous games: ● DRAW AND STUD POKER: These programs will keep your game sharp; ● NO-TRUMP BRIDGE: Develop your strategy and (hopefully) increase your skill. **(T1) Order No. 0063R $7.95.**

FLIGHT SIMULATIONS

RAMROM PATROL/TIE FIGHTER/KLINGON CAPTURE—•RAMROM PATROL: Destroy the RamRom ships before they capture you. •TIE FIGHTER: Wipe out the enemy Tie fighters and become a hero of the Rebellion. •KLINGON CAPTURE: You must capture the Klingon ship intact. **(T1) Order No. 0028R $7.95.**

FLIGHT PATH—This three-part package includes: •MOUNTAIN PILOT: Become a daring bush pilot and fly supplies to a remote mining camp. You must cross mountain ranges and struggle with headwinds, tricky navigation and rapidly diminishing fuel. •O'HARE: A control tower simulation for you would-be Air Traffice Controllers. You are responsible for the lives of hundreds of passengers as you guide aircraft through your control sector. •PRECISION APPROACH RADAR: Combines the skills of pilot and Air Traffic Controller, as your commands guide an aircraft in its approach to the field and a safe landing. **(T1) Order No. 0171R $9.95.**

BALL TURRET GUNNER—Imagine yourself at the control console of a strategic laser weapon, deep in the space lanes. Your hindsight detector informs you of a Gnat fighter coming in for an attack so you swivel you laser turret until you can see the target. Watch the Range Indicator and your Targeting Computer's readout closely, because you'll only have a fraction of a second to catch him in your sights. Will you transform the Gnat into a ball of ionized gas or will you see that blinding flash that means The Big Demotion? BALL TURRET GUNNER, with you choice of multiple levels of difficulty, optional sound effects and excellent graphics, is more than a game. It's an event to be savored. **(T1) Order No. 0051R $9.95.**

JET FIGHTER PILOT—In this brilliantly realistic simulation, you become the pilot of a twin turbo-jet fighter. Begin your mission from either the deck of a carrier or from an airfield. During flight, you'll need to constantly monitor your display and make the necessary adjustments to the throttle, flaps, and air spoilers; you must decide when to retract landing gear and release your drop tanks! There is an on-board Navigational Computer, a Glideslope/Localizer and a Weapons Control Computer. Earn your wings with JET FIGHTER PILOT. **(T1) Order No. 0159R $14.95.**

SPACE TREK II—Protect the quadrant from the invading Klingon warships. The Enterprise is equipped with phasers, photon torpedoes, impulse power and warp drive. **(T1) Order No. 0002R $7.95.**

AIR FLIGHT SIMULATION—Take off and land your aircraft without making a crater. This "instruments only" simulation starts you with a full tank of fuel, which gives you a maximum range of about 50 miles. You'll get constant updates of air speed, compass heading and altitude. After you've acquired a few hours of flight time, you can try flying a course against a map or doing aerobatic maneuvers. **T(1) Order No. 0017R. $9.95.**

SPACE TREK IV—•STELLAR WARS: Engage and destroy Tie fighters in your attack on the Death Star. For one player. •POPULATION SIMULATION: A two-player game where you control the economy of two neighboring planets. You must decide: Guns or Butter? **(T1) Order No. 0034R $7.95.**

BASIC AND INTERMEDIATE LUNAR LANDER—Bring your lander in under manual control. The basic version is for beginners; the intermediate version is more difficult, with a choice of landing areas and rugged terrain. **(T1) Order No. 0001R $7.95.**

COSMIC PATROL—We put you in command of a small interstellar patrol craft. You must defend Terran space and prey on the Quelon freighters that carry vital war supplies—but beware of their I-Fighter escorts. They're well armed, extremely fast and they NEVER miss! With its real-time action, impressive sound option and superb graphics, this machine-language program is the best of the genre. **(T1) Order No. 0223R $14.95.**

Airmail Pilot —Return to the early days of aviation. You must fly the mail from Columbus to Chicago. Your Jenny, a cloth-covered biplane, must take you through unpredictable winds, hail and electrical storms. Your mission is to get the mail through in the shortest possible time. There is an on-board clock to time you flight, from takeoff to touchdown. . . assuming you are able to complete it. **(T1) Order No. 0106R $9.95.**

NIGHT FLIGHT—Your mission is to fly ove the North Atlantic and make a nighttim photo/recon flight above the enemy fleet NIGHT FLIGHT lets you take-off, fly anc land a propellar-driven aircraft. You car practice approaches and landings with ar on-screen display of the landing field infor mation—it will practically teach you to fly **(T1) Order No. 0117R $9.95.**

COMP-U-NOVELS

WHO-DUN-IT? Criminal elements have committed five dastardly crimes. As the investigating detective, you must solve them.

You can compete against either Detective Nybbles, a computerized sleuth, or up to four other human detectives.

•DEDUCTION: Guess the order of four symbols out of six or seven different ones. To make things even more complicated, you can let the computer repeat symbols and have a range of 2401 possibilities. **(T1) Order No. 0047R $7.95.**

SANTA PARAVIA AND FIUMACCIO Become the ruler of a medieval city-state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen first. **(T1) Order No. 0043R $7.95.**

There are over 300 Instant Software dealers throughout the U.S.A and the world.

We ship the same day we receive your order.

CODE—Minimum System Required

(T1) = TRS-80 Model I Level II, 16K RAM
(T2) = TRS-80 Model I Level II, 16K RAM with Expansion Interface 16 + K RAM and one disk drive
(T3) = TRS-80 Model II, 32K RAM

*TRS-80 is a trademark of Tandy Corporation.

*A trademark of Tandy Corporation

SEE YOUR LOCAL INSTANT SOFTWARE DEALER OR

Just Call Toll-Free 1-800-258-5473

WRITE FOR OUR NEW INSTANT SOFTWARE CATALOG

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

Instant Software™

✓2

PETERBOROUGH, N.H. 03458

HOME/PERSONAL

THE WORDSLINGER—An economical word processing program that was designed for the individual user or small business featuring: automatic formatting; text editing; and tape storage. Once you've used the WORDSLINGER, you won't want to go back to your typewriter. **(T1) Order No. 0129R $29.95.**

MIMIC—Test your memory and reflexes with five versions of this popular game. You must match the sequence and location of symbols displayed on your monitor within the time limit. Instructions on how to produce accompanying sound effects. **(T1) Order No. 0066R $7.95**

CLIMATE COMP—This two-program package includes: WEATHER FORECASTER, which gives you a short range weather forecast based on the information that you enter and WEATHER PLOT, which will display climatological data for any major city in the United States. **(T1) Order No. 0102R-1 $19.95.**

BODY BUDDY—Includes these three programs: ● ADULT CALORIC REQUIREMENTS: Will determine your Basal Metabolic Rate and suggest strategies to achieve your ideal weight! ● FLEXI-DIET: Creates an "infinite" number of diet menus, on a day-to-day basis. Choose your caloric intake, from 600 to 2400 calories per day. The ● ANATOMY QUIZ program teaches a mini-lesson on the various organs of the human body, giving location, size and function(s). **(T1) Order No. 0109R $9.95.**

ENERGY CONSUMPTION—This program will record and analyze your utility bills for up to five years, when you supply the following information. Gas/Water/Electricity used and their respective costs. It will calculate six monthly usage averages and unit costs. Data can be compared for any month or multi-month periods. **(T1) Order No. 0132R $9.95.**

BUSINESS

SALES ANALYSIS—If your business is sales, you're faced with some unique problems. This package is divided into several modules to help solve those problems: The SALES ANALYSIS module is designed to provide guidelines for determining sales performance, to analyze this performance and show you where it can be improved. The DATA STORAGE module allows you to store data in an automated processing ledger. The MANAGEMENT ANALYSIS module can take all the sales records for your group and show you who your best salespersons are, who needs more training and give you a sales forecast. Finally, the MARKET ANALYSIS module can show you where determined sales efforts can produce the most success. **(T1) Order No. 0131R $24.95.**

ORACLE-80—will provide you with business analysis and forecasting capabilities previously available only on large computer and time-sharing systems. A flexible, professional time series analysis and forecasting package for use in product planning, business planning, sales forecasting and more. Financial managers and economists can analyze economic climates and investigate business cycles. ORACLE-80 is designed to be used and understood by the typical businessperson. All input and output is written in plain English and the package documentation carefully explains *all* the functions of the program. ORACLE-80 puts the future in your hands. **(T2) Order No. 0140R $75.00.**

BUSINESS PACKAGE IV—This business package contains two programs: ● BUSINESS CYCLE ANALYSIS: This program can plot the expansion and contraction cycles of any aspect of your business. ● FINANCIAL ANALYSIS: Now you can get the figures for any type of annuity, sinking fund, or mortgage and compute the yield and value for bonds. The package includes a blank data tape. **(T1) Order No. 0019R $9.95.**

FINANCIAL ASSISTANT—Compute the figures for a wide variety of business needs, including: ● DEPRECIATION: Figure depreciation on equipment five different ways. ● LOAN AMORTIZATION: Enter a few essential factors and get a complete breakdown of all costs and schedules of payment for any loan. ● FINANCIER: Performs thirteen common financial calculations. ● 1% FORECASTING: Use it to forecast sales, expenses, or any other historical data series. **(T2) Order No. 0072R $7.95.**

CHECK MANAGEMENT SYSTEM—Use this program for writing checks and maintaining records. You can make entries, edit/correct entries and print out the checks. It will also search and display records by number, code, date, description or amount. A Code and Search routine allows you to print a report of all checks written for specific expenses. You can print your letterhead and account number at the top of each report. System requirements: **(T2)** with a compatible tractor-feed printer. **0147RD $39.95.**

ACCOUNTS RECEIVABLE/ACCOUNTS PAYABLE—These Model I programs will handle the drudgery involved in AR/AP entries. They will also provide invoices, statements, reports and more. Each program is capable of handling up to 1500 entries per month, posted to as many as 760 accounts. The AR/AP package is ideal for any small business and can easily be used by anyone familiar with AR/AP operations. System requirements (in addition to **T2**: Three disk drives and a Line Printer (tractor-feed). **Order No. 0075RD $199.95.**

MAIL/LIST—With a five-inch drive, you can store up to 600 names per disk without DOS, or 300 names with DOS. The program maintains separate alphabetical and ZIP code files under constant sort. When you add a name or ZIP code to your list, it will be inserted into its correct position in the file. The program will record your data in nine fields: address, city, state, ZIP code, phone number, phone extension and name (2) plus a five character code field. The best feature of this program is the sort process that lets you determine alphabetical or ZIP code order for label printing. **(T2) Order No. 5000RD $99.00**

ONE-D MAILING LIST—A comprehensive mailing list program that will run on only ONE disk drive! Up to 17 fields of selection for name/address retrieval. Its features include: Auto-sort (alphabetic or ZIP code). Easy error correction and recovery. Prints selective listings. Supports up to 4 drives. Prints mailing labels and listing of all names on file. **(T2) Order No. 0123RD $24.95.**

EXECUTIVE EXPENSE REPORT GENERATOR—Provides you with emergency relief in the form of a clear, plausible expense layout. Input your grand total and cash advance (if any), and you'll receive an itemized expense report, from breakfast to snacks. **(T1) Order No. 0135R $9.95.**

GAMES

WINNER'S DELIGHT—Do you enjoy a challenge? Then try WINNER'S DELIGHT including: ● AMAZING: You must escape from a maze, one that you view from the inside, working against the clock; ● JUNIOR CHECKERS: Not your usual game of checkers . . . the challenge is to beat the computer in the fewest number of moves; ● JUMBO JIGSAW: Fit the pieces together in the fewest number of tries; ● THIRTEEN WAYS: Try to fill up your columns with the numbers you roll on the dice—the computer will try to fill its columns first! **(T1) Order No. 0124R $9.95.**

FUN PACKAGE I—Why call it "Fun Package"? Judge for yourself! This entertaining package includes: •ROCKET PILOT: Flying it is easy—it's the landing that's tough! •PAPER, ROCK, SCISSORS: It's the time-honored game just as you remember it, played against your TRS-80. •HEX I: Just when you master this puzzle game, the computer will increase the difficulty. •**MISSILE ATTACK: Use your missiles to protect your city from jet attack. Requires a TRS-80 Level I 16K. Order No. 0037R $7.95.**

DEMO III—The biggest package ISI has ever released, including: ● RACE 1: Careen around the race course as you try to beat the clock; ● TARGET UFO: Destroy all the invading UFOs; ● LIFE: Experiment with this simulation of the life cycle of a colony of bacteria; ● PHONE NUMBER CONVERTER: Change those hard to remember 7-digit phone numbers into easily remembered words; ● BIORHYTHM: Plot biorhythm curves for anyone, anytime; ● GRAPHICS PROGRAM: This program will show you what your TRS-80's graphics display can do; ● RACE 2: Five different tracks for the more experienced driver; ● HORSE RACE: Up to nine players can bet on and enjoy our most entertaining horse race program; ● DRAWING BOARD: Draw pictures or messages and store them in memory or on cassette tape with this easy-to-use program; ● 24-HOUR CLOCK: Transform your computer into an accurate digital clock. **(TI) Order No. 0055R $7.95**

OIL TYCOON—Avoid oil spills, blowouts and dry wells as you battle to become the world's richest oil tycoon. Two players become the owners of competing oil companies as they search for oil and control their companies. **(T1) Order No. 0023R $7.95.**

BOWLING—Let your TRS-80 set up the pins and keep score. One player can pick up spares and get strikes. **(T1) Order No. 0033R $7.95.**

DEMO II—contains: ● TIC-TAC-TOE: An old time favorite with three levels of difficulty; ● TIME TRIALS: Try to beat the clock as you race your car through curves, chutes, and chicanes; ● MAZE: One or two players can search through the maze for the secret square; ● HANGMAN: One or two players can try to guess the secret word; ● WHEEL OF FORTUNE: Choose your number, place your bet and see if you can break the bank (for one to eight players); ● HURRICANE: You can track and monitor hurricanes in *any* part of the world; ● BUGSY: Can you build your Z-80 bug before the computer does? ● HORSE RACE: Pick a sure winner and place your bet (for 1 to 100 players). **(T1) Order No. 0049R $7.95.**

BATTLEGROUND—It is late 1944 and the Allied forces are sweeping toward Berlin. As General in command, you study the map. At your command are tanks, planes, artillery, infantry, engineers, and vehicles. The battle map of your sector will fill with markers to show the development of your forces. You and your opponent will assume the roles of warring Generals, as the battle unfolds. The stark reality of World War II comes alive in BATTLEGROUND. **(T1) Order No. 0141R $9.95.**

SKIRMISH-80—Check out these great games: ● MISSION IMPOSSIBLE: Your objective in this real-time simulation is to drive your tank into a prison courtyard, rescue a jailed prisoner and escape; ● TRAP: A two-player game, in which you must maneuver your opponent into a position where he is hopelessly trapped; ● WIPEOUT: A two-player game in which your mobile gun gets points by destroying as many obstacles as possible, but be careful—some of those obstacles are explosive mines; ● BLOCK-'EM: A two-person competition in which your moving "snake" tries to force your opponent to hit either (1) your trail, (2) his own trail, (3) the boundaries of the field, or (4) any randomly place barriers. The strategy is, of course, to leave you opponent no safe move. **(T1) Order No. 0070R $9.95.**

POPULAR GAMES

GOLF/CROSS-OUT—Have fun with these exciting one-player games. Included are:
•GOLF: You won't need a mashie or putter—or a caddie, for that matter—to enjoy a challenging 18 holes.
•CROSS-OUT: Remove all but the center peg in this puzzle, and your neighbors will call you a genius. **(T1) Order No. 0009R $7.95.**

We ship the same day we receive your order.

*TRS-80 is a trademark of Tandy Corporation.

SEE YOUR LOCAL
INSTANT SOFTWARE DEALER OR

Just Call Toll-Free
1-800-258-5473

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

NO MATTER WHAT YOUR NEEDS ARE,
INSTANT SOFTWARE HAS
A PROGRAM FOR YOU

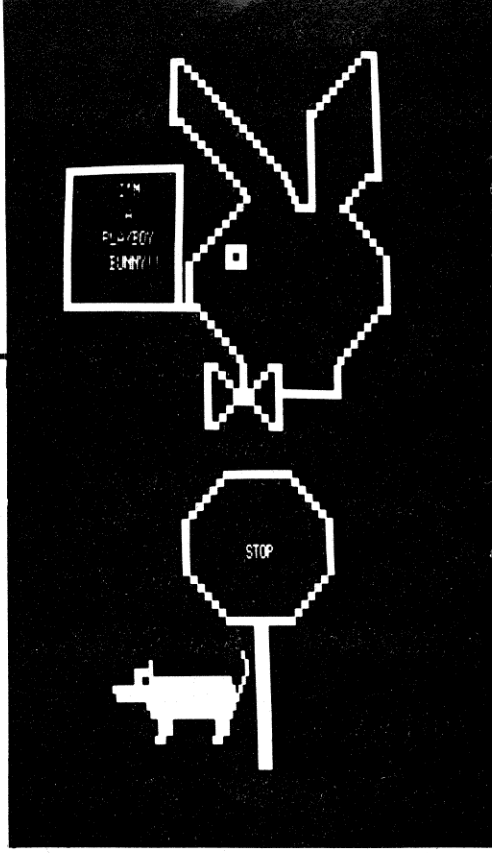
Instant Software™

✓2

PETERBOROUGH, N.H. 03458

Drawing the line at school.

Classroom Doodles



Ann Rosenberg
1303 Chimney Wood Drive
New Orleans, LA 70126

Graphics in the classroom not only teaches computer math but also promotes problem-solving, creativity and mental curiosity.

Stimulating a high school computer math class the week before the spring vacation was my main objective a few weeks ago. The students had been in the course for only eight weeks.

We had studied system commands such as NEW, LIST, DELETE, CSAVE, CLOAD and EDIT and program statements such as INPUT, FOR-NEXT, READ-DATA, INT, RND, IF THEN ELSE, GOSUB, and ON N GOTO. The group had successfully written several math oriented programs, but a change was now in order.

The class needed a project which was educational and fun. After rejecting several ideas, we decided on a graphics assignment.

Using Drawings

Each of the 12 students was given the following assignment:

> Using the TRS-80 Video Worksheet, draw a picture using horizontal, vertical and diagonal lines. From this, use SET(X,Y) and RESET(X,Y) to write the corresponding coding. After you have written your program, type it and debug it, and place your completed program on tape.

Their first reaction was "What should I draw?" Until now, the students had been given exact instructions on how and what their programs were to do. Now they seemed at a loss, but this changed quickly as they put their imaginations to work.

Before long, several programs were written and put to the test on the computer.

After looking at their graphic results, most of the students weren't satisfied with their simple stationary drawings. They went back to their worksheets to create more sophisticated ones.

The following are examples of what the students developed:

- Program Listing 1–Started out as a simple house but ended up a castle.
- Program Listing 2–Was a

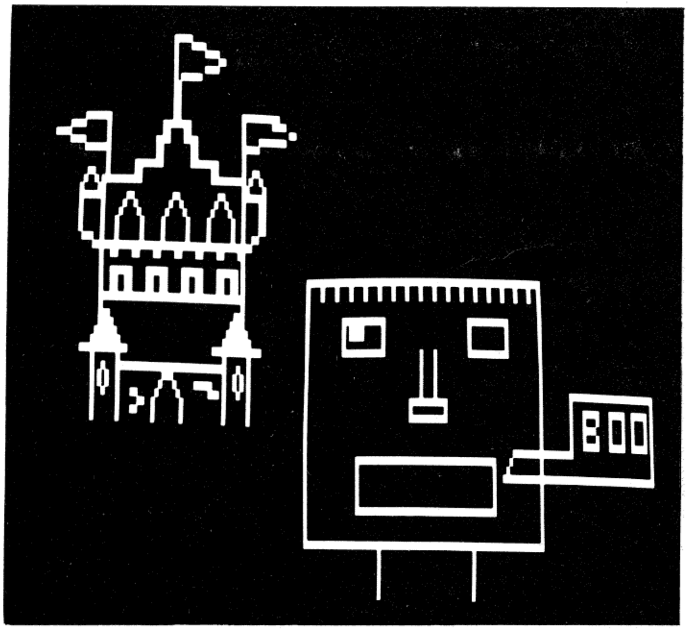
continues to p. 95

Program Listing 1. Castle

```
1 REM ******** CASTLE ********
2 REM PROGRAMMER: MICHAEL SHLENKER
5 CLS
1Ø FOR A=35 TO 79:SET(A,27):NEXT A
2Ø FOR B=37 TO 77:SET(B,33):NEXT B
3Ø FOR C=32 TO 44 :SET(C,39):NEXT C
4Ø FOR D=69 TO 82:SET(D,39):NEXT D
5Ø FOR E=43 TO 71:SET(E,41):NEXT E
6Ø FOR F=19 TO 35:SET(37,F):NEXT F
7Ø FOR G=17 TO 35:SET(77,G):NEXT G
8Ø SET(4Ø,28):SET(41,28):SET(46,28):SET(47,28):SET(52,2
    8):SET(53,28):SET(58,28):SET(59,28):SET(64,28):SET
    (65,28)
81 SET(7Ø,28):SET(71,28):SET(76,28)
9Ø FOR H=4Ø TO 47:SET(35,H):NEXT H
1ØØ FOR I=4Ø TO 47:SET(42,I):NEXT I
11Ø FOR J=4Ø TO 47:SET(72,J):NEXT J
12Ø FOR K=4Ø TO 47:SET(79,K):NEXT K
13Ø SET(38,35):SET(39,35)
14Ø FOR L=37 TO 4Ø:SET(L,36):NEXT L
15Ø FOR M=36 TO 41:SET(M,37):NEXT M
16Ø FOR N=35 TO 42:SET(N,38):NEXT N
17Ø SET(75,35):SET(76,35)
18Ø FOR O=74 TO 77:SET(O,36):NEXT O
19Ø FOR P=73 TO 78:SET(P,37):NEXT P
2ØØ FOR Q=72 TO 79:SET(Q,38):NEXT Q
21Ø SET(33,26):SET(34,26):SET(8Ø,26):SET(81,26)
22Ø FOR R=21TO25:SET(31,R):NEXT R
23Ø FOR S=21 TO 25:SET(32,R):NEXT S
24Ø FOR T=21 TO 25:SET(82,T):NEXT T
25Ø FOR U=21 TO 25:SET(83,U):NEXT U
26Ø FOR V=33 TO 36:SET(V,21):NEXT V
27Ø FOR W=78 TO 83:SET(W,21):NEXT W
28Ø FOR X=38 TO 47:SET(X,19):NEXT X
29Ø FOR Y=68 TO 76:SET(Y,19):NEXT Y
3ØØ FOR Z=46 TO  52:SET(Z,17):NEXT Z
31Ø FOR A=62 TO 69:SET(A,17):NEXT A
32Ø FOR B=52 TO 55:SET(B,14):NEXT B
33Ø FOR C=6Ø TO 63:SET(C,14):NEXT C
34Ø FOR D=54 TO 61:SET(D,12):NEXT D
35Ø SET(54,13):SET(55,13):SET(6Ø,13):SET(61,13)
36Ø FOR E=15 TO 16:SET(52,E):NEXT E
37Ø FOR F=15 TO 16:SET(53,F):NEXT F
38Ø FOR G=15 TO 16:SET(62,G):NEXT G
39Ø FORH=15 TO 16:SET(63,H):NEXT H
4ØØ SET(46,18):SET(47,18):SET(68,18):SET(69,18)
41Ø FOR I=3Ø TO 32:SET(4Ø,I):NEXT I
42Ø FOR J=3Ø TO 32:SET(41,J):NEXT J
43Ø FOR K=3Ø TO 32:SET(44,K):NEXT K
44Ø FOR L=3Ø TO 32:SET(45,L):NEXT L
45Ø FOR M=3Ø TO 32:SET(54,M):NEXT M
```

Program continues

```
460 FOR N=30 TO 32:SET(55,N):NEXT N
470 FOR O=30 TO 32:SET(60,O):NEXT O
480 FOR P=30 TO 32:SET(61,P):NEXT P
490 FOR Q=30 TO 32:SET(64,Q):NEXT Q
500 FOR R=30 TO 32:SET(65,R):NEXT R
510 FOR S=30 TO 32:SET(70,S):NEXT S
520 FOR T=30 TO 32:SET(71,T):NEXT T
530 FOR U=30 TO 32:SET(74,U):NEXT U
540 FOR V=30 TO 32:SET(75,V):NEXT V
550 SET(42,30):SET(43,30):SET(52,30):SET(53,30):SET(62,
    30):SET(63,30):SET(72,30):SET(73,30)
560 FOR W=24 TO 26:SET(41,W):NEXT W
570 FOR X=24 TO 26:SET(48,X):NEXT X
580 FOR Y=24 TO 26:SET(54,Y):NEXT Y
590 FOR Z=24 TO 26:SET(61,Z):NEXT Z
600 FOR A=24 TO 26:SET(67,A):NEXT A
610 FOR B=24 TO 26:SET(74,B):NEXT B
620 SET(42,43):SET(43,22):SET(44,21):SET(45,21):SET(46,
    22):SET(47,23)
630 SET(55,23):SET(56,22):SET(57,21):SET(58,21):SET(59,
    22):SET(60,23)
640 SET(68,23):SET(69,22):SET(70,21):SET(71,21):SET(72,
    22):SET(73,23)
650 SET(31,20):SET(32,19):SET(33,18):SET(34,18):SET(35,
    19):SET(36,20)
660 SET(78,20):SET(79,19):SET(80,18):SET(81,18):SET(82,
    19):SET(83,20)
670 FOR C=11 TO 16:SET(77,C):NEXT C
680 FOR D=11 TO 18:SET(38,D):NEXT D
690 FOR E=1 TO 11:SET(57,E):NEXT E
700 FOR F=7 TO 11:SET(58,F):NEXT F
710 FOR G=58 TO 61:SET(G,1):NEXT G
720 FOR H=62 TO 65:SET(H,2):NEXT H
730 FOR I=66 TO 69:SET(I,3):NEXT I
740 FOR J=70 TO 71:SET(J,4):NEXT J
750 FOR K=66 TO 69:SET(K,5):NEXT K
760 FOR L=59 TO 64:SET(L,6):NEXT L
770 FOR M=32 TO 37:SET(M,11):NEXT M
780 FOR N=28 TO 31:SET(N,12):NEXT N
790 FOR O=24 TO 27:SET(O,13):NEXT O
800 FOR P=28 TO 31:SET(P,14):NEXT P
810 FOR Q=32 TO 37:SET(Q,15):NEXT Q
820 FOR R=78 TO 87:SET(R,11):NEXT R
830 FOR S=86 TO 89:SET(S,12):NEXT S
840 FOR T=91 TO 92:SET(T,13):NEXT T
850 FOR U=82 TO 89:SET(U,14):NEXT U
860 FOR V=78 TO 81:SET(V,15):NEXT V
870 FOR W=45 TO 47:SET(52,W):NEXT W
880 FOR X=45 TO 47:SET(60,X):NEXT X
890 SET(53,44):SET(54,43):SET(55,42):SET(56,42):SET(57,
    42):SET(58,43):SET(59,44)
900 SET(38,41):SET(76,41):SET(38,44):SET(76,44)
910 FOR Y=42 TO 43:SET(37,Y):NEXT Y
920 FOR Z=42 TO 43:SET(39,Z):NEXT Z
930 FOR A=42 TO 43:SET(75,A):NEXT A
940 FOR B=42 TO 43:SET(77,B):NEXT B
960 SET(42,23)
970 FOR C=30 TO 32:SET(50,C):NEXT C
971 FOR D=30 TO 32:SET(51,D):NEXT D
980 SET(32,21):SET(53,17)
990 FOR E=64 TO 68:SET(E,43):NEXT E
1000 FOR F=68 TO 70:SET(F,44):NEXT F
1010 SET(46,44):SET(47,44):SET(48,45):SET(49,45):SET(46
     ,46):SET(47,46)
1020 GOTO 1020
```



Program Listing 2. Frankenstein

```
1 REM ******** FRANKENSTEIN ********
2 REM PROGRAMMER: GEORGE JANVIER (SOPHOMORE)
1Ø CLS
2Ø FOR X=24 TO 91
3Ø SET(X,9):SET(X,41)
4Ø NEXT X
5Ø FOR Y=9 TO 41
6Ø SET(24,Y):SET(91,Y)
65 NEXT Y
7Ø FOR Y=9 TO 11
8Ø FOR X=28 TO 88 STEP 4
9Ø SET(X,Y)
1ØØ NEXTX
11Ø NEXTY
12Ø FOR Y=14 TO 18
13Ø SET(34,Y):SET(45,Y):SET(7Ø,Y):SET(81,Y)
14Ø NEXTY
15Ø FOR X=34 TO 45
16Ø SET(X,14):SET(X,18)
17Ø NEXTX
18Ø FOR X=7Ø TO 81
19Ø SET(X,14):SET(X,18)
2ØØ NEXT X
21Ø FOR Y=18 TO 24
22Ø SET(56,Y):SET(6Ø,Y)
23Ø NEXTY
24Ø FOR Y=24 TO 26
25Ø SET(53,Y):SET(63,Y)
26Ø NEXTY
263 FOR X=53 TO 63
264 SET(X,24):SET(X,26)
265 NEXT X
27Ø FOR Y=41 TO 47
28Ø SET(44,Y):SET(71,Y)
29Ø NEXTY
3ØØ FOR X=38 TO 77
31Ø SET(X,31):SET(X,37)
32Ø NEXTX
33Ø FOR Y=31 TO 37
34Ø SET(38,Y):SET(77,Y)
35Ø NEXTY
36Ø FOR X=36 TO 39
37Ø FOR Y=15 TO 16
38Ø SET(X,Y)
39Ø NEXT Y
4ØØ NEXTX
4Ø1 FOR X=1ØØ TO 125
4Ø2 SET(X,23)
4Ø3 NEXT X
4Ø4 FOR X=8Ø TO 125
4Ø5 SET(X,33)
4Ø6 NEXTX
4Ø7 FOR Y=23 TO 33
4Ø8 SET(125,Y)
4Ø9 NEXT Y
41Ø FOR X=72 TO 75
42Ø FOR Y=15 TO 16
43Ø SET(X,Y)
44Ø NEXTY
445 NEXTX
446 FOR X=84 TO 1ØØ
447 SET(X,3Ø)
448 NEXTX
449 FOR Y=23 TO 3Ø
45Ø SET(1ØØ,Y)
451 NEXTY
452 FOR Y=25 TO 29
453 SET(1Ø4,Y):SET(1Ø8,Y):SET(112,Y):SET(116,Y):SET(119
    ,Y):SET(123,Y)
454 NEXTY
455 FOR X=1Ø4 TO 1Ø8
456 SET(X,25):SET(X,29):SET(X,27)
457 NEXTX
458 RESET(1Ø8,27)
459 FOR X=112TO116
46Ø SET(X,25):SET(X,29)
461 NEXTX
462 FOR X=119 TO 123
463 SET(X,25):SET(X,29)
464 NEXTX
465 SET(8Ø,33):SET(81,32):SET(82,31):SET(83,3Ø)
468 FOR X=1 TO 1ØØ:NEXTX
469 FOR X=38 TO 77
47Ø RESET(X,31):RESET(X,37)
48Ø NEXTX
49Ø FOR Y=31 TO 37
5ØØ RESET(38,Y):RESET(77,Y)
51Ø NEXTY
52Ø FOR X=36 TO 39
53Ø FOR Y=15 TO 16
54Ø RESET(X,Y)
55Ø NEXTY
56Ø NEXTX
57Ø FOR X=72 TO 75
58Ø FOR Y=15 TO 16
59Ø RESET(X,Y)
6ØØ NEXTY
61Ø NEXTX
62Ø FOR X=38 TO 77
```

```
63Ø SET(X,34)
64Ø NEXT X
65Ø FOR X=4Ø TO 43
66Ø FOR Y=16 TO 17
67Ø SET(X,Y)
68Ø NEXTY
69Ø NEXTX
7ØØ FOR X=76 TO 79
71Ø FOR Y=16 TO 17
72Ø SET(X,Y)
73Ø NEXTY
74Ø NEXTX
745 FOR X=1 TO1ØØ:NEXTX
75Ø FOR  X=38 TO 77
76Ø RESET(X,34)
77Ø NEXTX
78Ø FOR X=4Ø TO 43
79Ø FOR Y=16 TO 17
8ØØ RESET(X,Y)
81Ø NEXTY
82Ø NEXTX
83Ø FOR X=76 TO 79
84Ø FOR Y=16 TO 17
85Ø RESET(X,Y)
86Ø NEXTY
87Ø NEXTX
88Ø GOTO 3ØØ
```

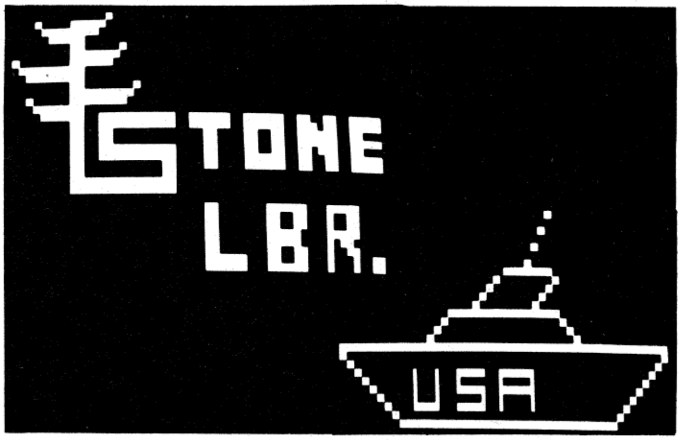
Program Listing 3. Playboy Bunny

```
1 REM ******** PLAYBOY BUNNY ********
2 REM PROGRAMMER: DANN SCHWARTZ (SOPHOMORE)
1Ø CLS
2Ø FOR N=Ø TO 8
3Ø SET(28,2+N)
4Ø SET(29,2+N)
5Ø NEXT N
6Ø  FOR R=Ø TO 9
7Ø SET(3Ø+2*R,1Ø+R)
8Ø SET(31+2*R,1Ø +R)
9Ø NEXT R
91 FOR Y=Ø TO 6
92 SET(3Ø+2*Y,2+Y)
93 SET(31+2*Y,2+Y)
94 NEXT Y
1ØØ FOR F=Ø TO 7
11Ø SET(3Ø+2*N,2+N)
12Ø SET(31+2*N,2+N)
13Ø NEXT F
14Ø SET(43,8)
15Ø FOR G=Ø TO 8
16Ø SET(44+2*G,9+G)
17Ø SET(45+2*G,9+G)
18Ø NEXT G
19Ø FOR T=Ø TO 4
2ØØ SET(6Ø+T,17+T)
21Ø SET(61+T,17+T)
22Ø NEXT T
23Ø SET(65,21)
24Ø SET(66,21)
245 SET(67,21)
25Ø FOR X=Ø TO 12
26Ø SET(68,9+X)
27Ø SET(69,9+X)
28Ø NEXT X
29Ø FOR B=Ø TO 8
3ØØ SET(85-2*B,1+B)
31Ø SET(86-2*B,1+B)
32Ø NEXT B
33Ø FOR C=Ø TO 15
34Ø SET(88,2+C)
35Ø SET(89,2+C)
36Ø NEXT C
361 SET(88,1):SET(87,1):SET(89,1)
365 SET(87,17)
37Ø FOR D=Ø TO 4
38Ø SET(85-2*D,17+D)
39Ø SET(86-2*D,17+D)
4ØØ NEXT D
41Ø FOR E=Ø TO 6
42Ø SET(49-2*E,2Ø+E)
43Ø SET(5Ø-2*E,2Ø+E)
44Ø SET(78+2*E,21+E)
45Ø SET(79+2*E,21+E)
46Ø NEXT E
465 SET(35,27):SET(36,27)
47Ø FOR F=Ø TO 5
48Ø SET(33,28+F)
49Ø SET(34,28+F)
5ØØ SET(9Ø,28+F)
51Ø SET(91,28+F)
52Ø NEXT F
53Ø FOR H=Ø TO 6
```


✓Reader Service—see page 242

```
540 SET(35+2*H,33+H)
550 SET(36+2*H,33+H)
560 SET(89-2*H,33+H)
570 SET(88-2*H,33+H)
580 NEXT H
590 FOR I=0 TO 2
600 SET(48 ,40+I)
610 SET(49,40+I)
620 SET(75,40+I)
630 SET(76,40+I)
640 NEXT I
650 FOR J=0 TO 15
660 SET(60+J,43)
670 SET(61+J,43)
680 NEXT J
690 FOR K=0 TO 7
700 SET(38,40+K)
710 SET(39,40+K)
720 SET(58,40+K)
730 SET(59,40+K)
740 NEXT K
750 FOR Z=0 TO 3
760 SET(40+2*Z,40+Z)
770 SET(41+2*Z,40+Z)
780 SET(50+2*Z,43-Z)
790 SET(51+2*Z,43-Z)
800 SET(50+2*Z,44+Z)
810 SET(51+2*Z,44+Z)
820 SET(40+2*Z,47-Z)
830 SET(41+2*Z,47-Z)
840 NEXT Z
845 FOR K=16 TO 32
850 SET(30,K):SET(31,K)
855 NEXT K
860 SET(49,44):SET(48,43):SET(49,43)
870 FOR L=0 TO 5
880 SET(44+L,26):SET(44+L,27):SET(44+L,28)
890 NEXT L
891 SET(48,44)
950 PRINT @384,TAB(7);"I'M";
955 PRINT @ 448, TAB(8);"A";
960 PRINT @512, TAB(5);"PLAYBOY";
970 PRINT @ 576, TAB(6);"BUNNY!!";
973 FOR M=0 TO 32
975 SET(0+M,33)
977 NEXT M
980 FOR B=16 TO 32

983 SET(0,B):SET(1,B)
985 NEXT B
996 FOR S=0 TO 30
997 SET(S,16)
999 NEXT S
1100 SET(47,27):SET(46,27)
1120 RESET(47,27):RESET(46,27)
1130 GOTO 1100
5000 GOTO 5000
```



Program Listing 4. Ship

```
1 REM ******** SHIP ********
2 REM PROGRAMMER: DAVID FUCHS (SOPHOMORE)
5 CLS
10 FOR X = 34 TO 95: SET(X,44):NEXT X
20 FOR X = 18 TO 111:SET(X ,35):NEXT X
30 FOR X = 64 TO 77:SET(X,26):NEXT X
40 FOR X = 70 TO 71:SET(X,25):NEXT X
50 FOR X = 72 TO 73:SET(X,23):NEXT X
```

```
60 FOR X = 74 TO 75:SET (X,21):NEXT X
70 FOR X = 76 TO 77:SET(X,19):NEXT X
130 FOR X = 0 TO 7:SET(18+2*X,36+X):SET(19 +2* X,36+X):
     SET(111-2*X,36+X):SET(110-2*X,36 +X):NEXT X
140 FOR X= 0 TO 2:SET(47-2*X,32+X):SET(46-2*X,32+X):SET
     (80+2*X,32+X):SET(81+2*X,32+X):NEXT X
150 FOR X =0TO 31:SET(48+X,31):NEXT X
160 FOR X = 0 TO 3:SET(62-2*X,27+X):SET(61 - 2 * X,27 +
     X):SET(78-2*X,27+X):SET(79-2*X,27+X):NEXT X
170 FOR Y= 38 TO 42:SET(38,Y):NEXT Y
180 FOR Y = 38 TO 42:SET(44,Y):NEXT Y
190 FOR X = 38 TO 44:SET(X,42):NEXT X
200 FOR X = 50 TO 57 :SET (X,38):NEXT X
210 FOR X=50 TO 57:SET(X,40):NEXT X
220 FOR X = 50 TO 57:SET(X,42):NEXT X
230 FOR Y =38 TO 40:SET(50,Y):NEXT Y
240 FOR Y = 40 TO 42 :SET(57,Y):NEXT Y
250 FOR X = 62 TO 71:SET (X,38):NEXT X
260 FOR X= 62 TO 71:SET(X,40):NEXT X
270 FOR Y = 38 TO 42 :SET(62,Y):NEXT Y
280 FOR Y = 38 TO 42:SET(71,Y):NEXT Y
290 FOR Y = 38 TO 42:RESET(38,Y):NEXT Y
300 FOR Y = 38 TO 42:RESET(44,Y):NEXT Y
310 FOR X = 38 TO 44:RESET(X,42):NEXT X
320 FOR X = 50 TO 57:RESET(X ,38):NEXT X
330 FOR X = 50 TO 57:RESET(X,40):NEXT X
340 FOR X = 50 TO 57:RESET(X,42):NEXT X
350 FOR Y =38 TO 40:RESET(50,Y):NEXT Y
360 FOR Y =40 TO 42:RESET(57,Y):NEXT Y
370 FOR X = 62 TO 71:RESET(X,38):NEXT X
380 FOR X=62 TO 71:RESET(X,40):NEXT X
390 FOR Y = 38 TO 42:RESET(62,Y):NEXT Y
400 FOR Y=38 TO 42:RESET(71,Y):NEXT Y
410 GOTO 170
5000 GOTO5000
```

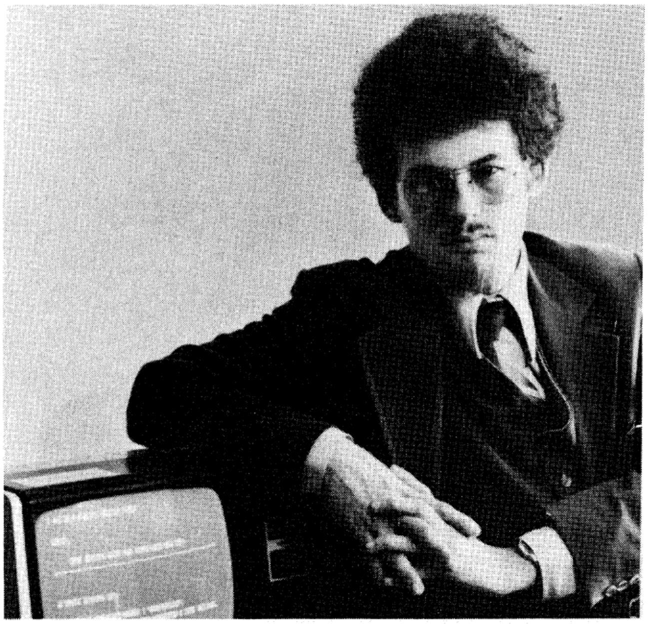
Program Listing 5. Dog

```
1 REM ******** DOG ********
2 REM PROGRAMMER: RANDY KESSLER (SENIOR)
10 CLS
20 FORX=54TO73
30 Y=12
40 SET(X,Y)
50 NEXT X
60 SET(52,13)
70 SET(53,13)
80 SET(51,14)
90 SET(50,14)
100 SET(49,15)
110 SET(48,15)
120 SET(47,16)
130 SET(46,16)
140 SET(45,17)
150 SET(44,17)
160 SET(43,18)
170 SET(42,18)
175 X=74
180 FORY=13TO18
190 FORD=1TO2
200 SET(X,Y)
210 X=X+1
220 NEXTD
230 NEXTY
240 FORX=42TO43
250 FORY=19TO24
260 SET(X,Y)
270 NEXTY
280 NEXTX
290 FORX=84TO85
300 FORY=19TO24
310 SET(X,Y)
320 NEXTY
330 NEXTX
340 X=44
350 FORY=25TO30
360 FORD=1TO2
370 SET(X,Y)
375 X=X+1
380 NEXTD
390 NEXTY
400 FORX=54TO73
410 Y=30
420 SET(X,Y)
430 NEXTX
435 X=83
440 FORY=25TO29
450 FORD=1TO2
460 SET(X,Y)
470 X=X-1
480 NEXTD
490 NEXTY
500 FORY=31TO47
```

```
51Ø FORX=62TO65
52Ø SET(X,Y)
53Ø NEXTX
54Ø NEXTY
56Ø PRINT@478,"STOP";
6ØØ SET(58,34)
61Ø SET(59,35)
62Ø SET(59,36)
63Ø SET(58,37)
64Ø SET(57,38)
65Ø SET(56,39)
66Ø FORY=38TO4Ø
67Ø X=55
68Ø SET(X,Y)
69Ø NEXTY
7ØØ FORY=37TO41
71Ø FORX=32TO54
72Ø SET(X,Y)
73Ø NEXTX
74Ø NEXT Y
75Ø SET(32,35)
76Ø SET(32,36)
77Ø FORX=28TO32
78Ø Y=36
79Ø SET(X,Y)
8ØØ NEXTX
81Ø SET(28,37)
82Ø SET(29,37)
83Ø FORX=23TO31
84Ø FORY=38TO39
85Ø SET(X,Y)
86Ø NEXTY
87Ø NEXTX
88Ø SET(22,38)
89Ø SET(31,4Ø)
9ØØ FORX=34TO37
91Ø FORY=42TO43
92Ø SET(X,Y)
93Ø NEXTY
94Ø NEXTX
95Ø FORX=33TO35
96Ø Y=44
97Ø SET(X,Y)
98Ø NEXTX
99Ø FORX=5ØTO53
1ØØØ FORY=42TO43
1Ø1Ø SET(X,Y)
1Ø2Ø NEXTY
1Ø3Ø NEXTX
1Ø4Ø SET(49,44)
1Ø5Ø SET(5Ø,44)
1Ø6Ø SET(51,44)
1Ø7Ø FORX=1TO5ØØ:NEXTX
1Ø71 FORK=62TO57STEP-1
1Ø73 SET(K,44)
1Ø75 J=1
1Ø8Ø SET(56,4Ø)
1Ø9Ø FORX=1TO6Ø:NEXTX
11ØØ SET(57,41)
111Ø FORX=1TO1ØØ:NEXTX
112Ø RESET(56,4Ø)
113Ø SET(58,42)
114Ø FORX=1TO1ØØ:NEXTX
115Ø RESET(57,41)
116Ø SET(59,43)
117Ø FORX=1TO1ØØ:NEXTX
12ØØ RESET(59,43)
121Ø RESET(58,42)
1212 SET(K,44)
1215 PRINT@95Ø,"PLOP!";
1217 FORX=1TO2ØØ:NEXTX
1218 PRINT@95Ø,"      ";
1225 NEXTK
123Ø SET(59,44)
124Ø PRINT@945,"WHAT A MESS!";
125Ø PRINT@478,"PHEW ";
2222 GOTO2222
```

Program Listing 6. Stone Lumber Logo

```
1 REM ******** STONE LUMBER ********
2 REM PROGRAMMER: BOB STONE (SENIOR)
1Ø CLS
2Ø SET (42,4):SET (43,4)
3Ø FOR Y=5 TO 26
4Ø SET (32,Y):SET(33,Y):SET(34,Y):SET (35,Y)
5Ø NEXT Y
6Ø FOR Y=17 TO 22
7Ø SET (4Ø,Y):SET(41,Y):SET (42,Y):SET (43,Y)
8Ø NEXT Y
9Ø FOR Y=21 TO 26
1ØØ SET (58,Y):SET (59,Y):SET (6Ø,Y):SET (61,Y)
11Ø NEXT Y
12Ø FOR Y=17 TO 23
13Ø SET(7Ø,Y):SET(71,Y):SET(72,Y):SET(73,Y):SET(82,Y):S
    ET(83,Y):SET(84,Y):SET(85,Y)
```



Adventure INTERNATIONAL

A DIVISION OF SCOTT ADAMS, INC.

Scott Adams
ADVENTURE SERIES

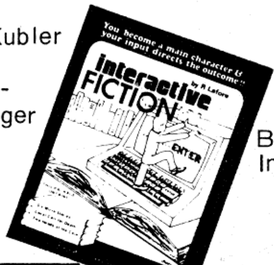
WE WANT TO PUBLISH YOUR SOFTWARE!

TRS-80 Models 1,2,3,Color and Pocket, Apple and Atari Computers.

JOIN THE LEADING AUTHORS IN THE COMPUTER INDUSTRY THAT HAVE THEIR SOFTWARE PUBLISHED BY ADVENTURE INTERNATIONAL.

Doug Carlston
Galactic Series

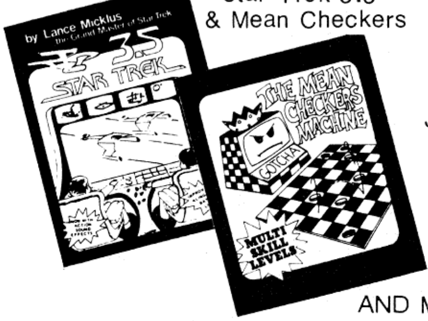
Dale Kubler
Maxi Micro-Manager



James Talley
Kid-Ventures

Bob Lafore
Interactive Fiction

Lance Micklus
Star Trek 3.5
& Mean Checkers



Jack Moncrief
& Mike Wall
Lunar Lander

Chameleon Software
Maces and Magic Series

AND MANY MORE!

Silver Flash
Pinball
Southern Cross Systems

We offer excellent Royalties, thorough advertising, great Dealer exposure, professional packaging and support! WE CARE That's why we are one of the leaders in the industry! To submit a program, send a machine readable copy along with all the necessary documentation to:

ADVENTURE INTERNATIONAL ✓97
SOFTWARE REVIEW BOARD
P. O. BOX 729
CASSELBERRY, FL 32707

A postcard will be sent to you when your submission arrives to verify receipt. The Review Board takes from one to three weeks to thoroughly review your program.

Missile Attack
Cornsoft Group

SEND YOUR SUBMISSION TODAY!

```
140 SET(90,Y):SET(91,Y):SET(92,Y):SET(93,Y):SET(98,Y):S
    ET(99,Y):SET(100,Y):SET(101,Y)
145 SET(106,Y):SET(107,Y):SET(108,Y):SET(109,Y):SET(114
    ,Y):SET(115,Y):SET(116,Y):SET(117,Y)
150 NEXT Y
160 FOR X=32 TO 41
170 SET (X,5)
180 NEXT X
190 FOR X=24 TO 39
200 SET(X,6)
210 NEXT X
220 SET (22,5):SET (23,5)
230 FOR X=26 TO 31
240 SET (X,7)
250 NEXT X
260 FOR X=36 TO 45
270 SET (X,9)
280 NEXT X
290 FOR X=36 TO 43
300 SET (X,10)
310 NEXT X
320 FOR X=18 TO 31
330 SET (X,11)
340 NEXT X
350 FOR X=20 TO 31
360 SET (X,12)
370 NEXT X
380 FOR X=36 TO 49
390 SET (X,14)
400 NEXT X
410 FOR X=36 TO 47
420 SET (X,15)
430 NEXT X
440 SET (16,10):SET (17,10)
450 FOR X=22 TO 31
460 SET (X,16)
470 NEXT X
480 FOR X=24 TO 31
490 SET (X,17)
500 NEXT X
510 SET (46,8):SET (47,8)
520 SET (50,13):SET (51,13)
530 SET (20,15):SET (21,15)
540 FOR X=44 TO 61
550 SET (X,17):SET (X,18)
560 NEXT X
570 FOR X=44 TO 57
580 SET (X,21):SET (X,22)
590 NEXT X
600 FOR X=36 TO 57
610 SET (X,25):SET (X,26)
620 NEXT X
630 FOR X=66 TO 77
640 SET (X,17):SET (X,18)
650 NEXT X
660 FOR X=86 TO 89
670 SET (X,17): SET (X,18):SET (X,22): SET (X,23)
680 NEXT X
690 FOR X=102 TO 103
700 SET (X,18):SET (X,19):SET (X,20)
710 NEXT X
720 FOR X=104 TO 105
730 SET (X,19):SET (X,20):SET (X,21)
740 NEXT X
750 FOR X=118 TO 123
760 SET (X,17):SET (X,18)
770 SET (X,20)
780 SET (X,22):SET (X,23)
790 NEXT X
800 FOR Y=28 TO 35
810 SET (70,Y):SET (71,Y):SET (72,Y):SET (73,Y)
820 SET (88,Y):SET (89,Y):SET (90,Y):SET (91,Y)
825 SET (96,Y):SET(97,Y)
830 SET (106,Y):SET (107,Y)
840 NEXT Y
850 FOR X=74 TO 81
860 SET (X,34):SET (X,35)
870 NEXT X
880 FOR X=92 TO 99
890 SET (X,28):SET (X,35)
900 NEXT X
910 FOR Y=31 TO 32
920 SET (92,Y):SET (93,Y):SET (94,Y):SET (95,Y)
930 NEXT Y
940 FOR Y =29 TO 30
950 SET (98,Y):SET (99,Y)
960 NEXT Y
970 FOR Y=33 TO 34
980 SET(98,Y):SET (99,Y)
990 NEXT Y
1000 SET (108,28):SET (109,28):SET (110,28):SET (111,28
     )
1010 SET (108,31):SET (109,31)
1020 FOR Y=28 TO 33
1030 SET (112,Y):SET (113,Y)
1040 NEXT Y
1050 FOR Y=28 TO 31
1060 SET (114,Y):SET (115,Y)
1070 NEXT Y
1080 FOR Y=31 TO 32
1090 SET (110,Y):SET (111,Y)
1100 NEXT Y
1110 FOR Y=33 TO 35
1120 SET (114,Y):SET (115,Y)
1130 NEXT Y
1140 FOR Y=34 TO 35
1150 SET (120,Y):SET (121,Y):SET (122,Y):SET (123,Y)
1160 NEXT Y
1200 GOTO 1200
```

strange face but became a Frankenstein, complete with moving lips and shifting eyes.

- Program Listing 3-Was an Easter Bunny but quickly transformed into a Playboy Bunny with blinking eyes.
- Program Listing 4-Was a plain and simple boat until the letters USA were added and appeared to move across the body of the ship.
- Program Listing 5-Is the class favorite. The first day it was a simple stop sign. The second day, the student added a dog. With a little prodding from classmates, the student had the dog add a few "plops" at the base of the sign.
- Program Listing 6-Is the advertising logo for a lumber company owned by a student's father.

By the end of the week, everyone had completed exciting and creative graphic displays. Each was so proud of his/her accomplishments, that it was not unusual for friends and teachers to stop by the computer room and view the drawings.

The students not only enjoyed this assignment, but they became proficient at using graphics. Future assignments will be much easier for them.

Instead of just solving right triangles with the Pythagorean Theorem or general triangles with the Law of Sines and Cosines, they will be able to draw these triangles.

The graphics can also be used to make bar graphs, display data and write games.■

Hints and kinks from the land of institutionalized 80s.

Notes From the Classroom

Ralph L. von Kaenel
2110 N.W. Couch
Camas, WA 98607

If any of you educators hope to introduce your students to the computer world for their every day use or a career, this article may help you avoid some classroom stumbling blocks.

Mistakes

During the past year, we've given five TRS-80 Level IIs a complete run through and have made as many mistakes and discoveries as possible.

Our first mistake was to place the computers back to back, leading to an undesirable distortion. So, we placed them side by side. Even when doing this, we had to leave about ten inches between units. Our wiring was a jumbled mess of extension cords. By putting each computer on its own bar, preferably with isolated filters, we solved the problem. When wires are crossed from one computer to the next, an occasional 'glitch' will appear. There is enough stray static around without creating some of your own.

We also had a rash of spoiled tapes during a period of dry windy weather. We realized then that some students seemed to be full of static. The electrical kind as well as the verbal. By having the students touch a water faucet before touching their tapes, this seemed to help.

The next big problem was with our tape recorders. With four CTR-80 recorders and one CTR-41, the recorders ran the tapes at the same speed. The CTR-41 counter, however, rotated at a much faster rate. If a student recorded on a CTR-80, for example, and marked the footage, then moved to the CTR-41, he would, in effect, tape over a program due to the difference in speed in recorded footage. We exchanged the 41 for another 80 and that completely solved it.

When you spend hours producing a program you realize quickly the common sense in locating an inexpensive method for reproducing from your master. We found that a jumper cable between two CTR-80s, with the volume down to level two for recording and level four for loading, provided perfect reproduction. We disconnected the speaker on one recorder to stop outside noise from interfering. This allowed students to exchange programs without tying up a computer and also work without creating any disturbances.

One recorder did give us some trouble until we read the article by John Victor in the February issue of *80 Microcomputing* about misaligned heads on tape recorders. One keyboard also needed a modification. This was furnished free of charge and returned in three days.

Adjusting

With beginners there are many false starts and changes to programs, so we found a bulk eraser a necessity.

It's indispensible and less expensive than replacement tapes.

During the hot weather, we installed fans in the classroom to help cool the computers. We discovered that the computers worked best at 60 to 85 degrees. Anything above 85 caused bad tape loads.

During the winter prolonged periods of cold weather had the same effect. So, we plugged in an electric heater to hold the temperature at 60 overnight—and solved another troublesome problem.

For those of you, who still own a CTR-41 tape recorder, hang onto it.

The new CTR-80 which comes with the newer computers is a handy item but it ruins tapes if it is turned off or on during a loading run. If you make some simple modifications (see the April issue of *80 Micro-*

Beginning computer students at Camas High School, Camas, WA, learn to use 80s.

computing, page 110) you'll be more than satisfied with the CTR-41.

If you want to include a switch in the speaker circuit (Fig. 2, April, 1980 article), it will enable you to turn the audio off after locating the program start. The audio permits you to record the footage and be aware of the end of a program, so you won't have to guess or waste good computing time.

Each student has an hour when he or she may use the computers. Thus, I needed a switch on each computer. With nothing available locally, I settled for a plug-in. By cutting the power line on the power supply unit and installing a six-prong plug-in 274-207 and in-line 274-208, I keyed each computer and numbered it. See Fig. 1 for a simple method to connect plug-ins for up to nine computers. I also have a master plug-in that gives me control over any of the computers. The plug-ins are made so that they cannot be plugged in except one way, which prevents a possible short.

We are teaching programming without the use of software. Our students are juniors and seniors at the high school level and are generally business and accounting majors. The first quarter is spent in the textbook, and the second quarter is spent with the *Level II Reference Manual.* During the second quarter, the students work on programming in areas of interest.

During our first year, we converted the 16K Level II to Level I, which used software that would be compatible with the textbooks available. Now that author David A. Lien is marketing another book, *Learning Level II*, (which upgrades the original text to Level II) we thankfully say goodbye to conversion tapes. We now train the beginning student on Level II.

We are busily converting our remaining texts to Level II and hoping for an easier year.

The two sets of prepared training tapes, Level II Basic Course, parts I and II, are beautifully done and easily understood. Students can read through them in just a few hours. Sitting quietly at a computer and pressing keys with one finger, there is a tendency to rush through this without practicing the many little programs shown as learning applications. I've found them more impressive as review or advanced work. The student will then take time to make notes or by then has lost his fear of breaking into the middle of a lesson to practice an example.

A fine book to have in your library is Radio Shack's unabridged *Dictionary of Electronics.* And another invaluable source of information is your local computer club.

If you do indeed get computers for your class, you can expect some long and interesting hours. The serious student will take the chance to confront challenges presented and create some new ones for you. ■

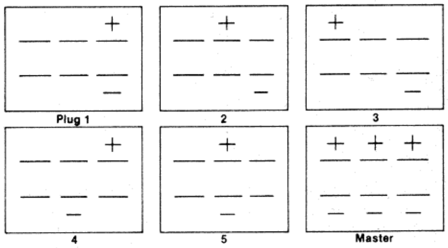


Fig. 1. Soldering the plug-in 274-207 and in-line 274-208 as keys to the power unit. Make sure they are connected to the same poles are marked, or there will be an open circuit.

POSTMAN DATA HANDLER

Ver. 1.0 – by Fred LaForest

A **machine language** mailing list program that will do:

- 650 Labels on a 35 track disk drive
- 1534 lables on an 80 track drive
- 10 fields (2 user defined)
- FAST SORTS 500 records in 30 seconds (use any or all keys in any order)
- Fully usable on a one (1) drive system (capacities shown are for a single drive system)
- Any label stock ½" thru 1½" vertical (single label horizontal)
- Print one label or a sequence of labels
- **Purge duplicates** with or without user assistance
- 9 digit zip code
- Fast search on any field - random access - 3 second average
- Easy screen for fast editing
- REQUIRES MIN. 1 DRIVE and 32K OF MEMORY, TRS-80*, MOD I

This program is now available in 2 different packages.

1) A **sample package** that does all the functions of the full system (except the purge) and sells for $25 and is to be used as a sales tool only. This is a fully operational package but can not be enlarged or modified in any way. Comes with the complete documentation and credit can be issued to the real package if returned to its place of purchase within 20 days.

2) The **full program** that includes the PURGE function with full documentation. This package will be updated as time goes on with new ideas so it includes a registration card.

Now it even works on Newdos-80

List Price $125.00

Send $25 for Sample Package – if not everything you expected, return sample disk for full refund (less shipping). You can't lose!!

THE CREATOR

The CREATOR is a new type of program for the micro-computer operator. Yes operator!! Easy enough for the person just getting into the market. Use and create a program that is very sophisticated that programmers will comment highly about. The program will create error free basic programming code. Not almost ready to run BUT READY TO RUN WHEN YOU ARE FINISHED. YES, gives birth to a program. Just answer simple questions and have a simple background in the disk system of your computer (if you read your basic manual when you have questions you will have no problems). THIS PROGRAM IS NOT A DATA BASE!!

Now in the package comes the report generator that is in the same concept as the CREATOR. It is called REPORTER. This program creates report output for the CREATOR for either screen or printer.

These 2 programs are on one diskette and are available for only $295 complete. The system requirements are one of the below:

TRS-80 MOD I, 32K DISK
TRS-80 MOD II, 64K DISK
APPLE II, 32K DISK

This is the most outstanding programming package available from anywhere. Now you can create INVENTORY SYSTEMS, PAYABLES and RECEIVABLES, CHECK REGISTER and EXPENSE REGISTER, and MUCH MORE!

This package is ready for delivery only $295 for any one of the systems above (PLEASE STATE SYSTEM WHEN ORDERING).

10% OFF Save 10% this month Software Only!!

SUPER-UTILITY

by K. Watt

— MAIN PROGRAM LIST —

ZAP UTILITY
Display Sector (Disk, File)
Display Memory
Compare Disk Sectors
Copy Disk Sectors
Verify Disk Sectors
Zero Disk Sectors
String Search
Sector Search

PURGE UTILITY
Kill Selected Files
Get Disk Directory
Zero Unused Directory Entries
Zero Unused Granules
Remove System Files
Kill By Category
Change Name, Date, Password, Auto Command
Change File Paramaters
Remove Passwords

DISK FORMAT UTILITY
Standard Format
Format Without Erase
Special Format
Read Address Marks

DISK COPY UTILITY
Standard Copy With Format
Standard Copy Without Format
Special Copy (to back up any protected disks) Purchaser Use - Only for his own personal disks.

TAPE COPY UTILITY
This program is to make back-up of any TRS-80 tape, no matter how it is recorded (note again this program is for the use of the original purchaser for his own programs only)

DISK REPAIR UTILITY
Repair Gat Table
Repair Hit Table
Repair Boot
Read Protect Directory Track
Recover Killed Files
Check Directory

MEMORY UTILITY
Move Memory
Exchange Memory
Compare Memory
Zero Memory
Test Memory
Input Byte From Port
Output Byte To Port
Memory To Disk
Disk To Memory

— For TRS-80, MOD I —

For a more complete overview, send a self addressed stamped envelope. This program is sold on disk only and retails for $49.95.

OTHER THINGS OF INTEREST

For The TRS-80* MOD I

ST-80 III the top of the line in communication packages, MOD I.

List	Cash Discount
$150.00	$137.00

LEX-II MODEM answer/orig.

List	Cash Discount
$179.00	$139.00

SINGLE 5¼ DISK DRIVES 40 Track with Power Supply and Case.

List	Cash Discount
$399.00	$299.00

DUAL 5¼ DISK DRIVES 40 Track with Power Supply and Case.

List	Cash Discount
$799.00	$579.00

80 TRACK DRIVES with Power Supplv and Case.

List	Cash Discount
$650.00	$410.00

DUAL DRIVES 80 Track with Power Supply and Case.

List	Cash Discount
$1100.00	$800.00

All hardware must be pre-paid to receive cash discount
or
Use your Master Card and Visa

Dealer Inquiries Invited. SSM SOFT SECTOR MARKETING, INCORPORATED ✓434 **Dealer Inquiries Invited.**

6250 Middlebelt • Garden City, MI 48135 • 1 (313) 425-4020

C.O.D. – Certified Check, M.O. or Cash only. Sorry no C.O.D. over $150.00! Most orders shipped next day. All orders must have shipping included. Please add 2% or $2.50, which ever is higher for shipping. Michigan residents, please add 4% tax. Add extra $1.50 for C.O.D. Personal checks take 3 weeks to clear. Send $1.00 for catalog - get $2.00 credit on next order.

✓Reader Service—see page 242

A subroutine by any other name.

Into the 80's

I. R. Sinclair
89 Alexandra Road
Sible Hedinghan
Halstead, Essex CO93N
England

Once your programs pass the very simple stage, you'll need to present a menu. As the word suggests, a menu is a list of choices for the user. He makes a choice. The way it is presented and the way the user makes the choice are all the difference in the world between a program that is a joy and one which is a pain.

The menu should, first, give the user some idea of what the choice is—not just a listing of five numbers! The description needn't elaborate, none of your "sun-ripened section of choice, West Coast subroutine, delectably preserved in quotes"—it isn't that sort of menu—but it must tell all. Fig.1 shows a typical short menu, with choices for keyboard entry, entry of data from cassette, and termination.

Termination is important, if the menu is presented several times in the program. There's nothing as infuriating as having to go through several unnecessary steps just to stop a program. The BREAK key can be used to terminate, but it makes sense to construct programs that need little interference.

Using the command PRINT CHR$(23) just before the menu printout prints it in double-sized characters. The character size can then be returned to normal later with a PRINTCHR$(28), POKE 16445,0, or CLS command.

The simplest way to carry out a menu choice is to type and enter the number of the chosen item. In ordinary BASIC, this would be done as shown in Fig. 2. The choice is made by typing one of the numbers shown in the menu and ENTERing.

Line 30 is an error trap: If you have selected a number that doesn't exist, you are informed, and steered back to the menu to try again. That's an important point. If an error trap causes a return with no explanation, the user may not know that there is an error, because there is only a slight flicker on the screen. Showing a message leaves the user knowing that there has been an incorrect entry.

Lines 40 through 70 implement the choice. For each possible menu number, the program is instructed to jump to a different line, or to end. At each of the new lines (the examples show 300,700,1000), a new section of program must start. This will carry out the action promised by the menu.

That's how a Brand X computer might deal with a menu, but the TRS-80 has a whole lot of tricks up its sleeve. One of the tricks, as far as a menu choice is concerned, is the command ON K GOTO. . . .

Fig. 3 shows this replacing line 40 in Fig. 2, with lines 50 through 70 deleted and a new line 5000 added. When you enter a number, it is assigned the variable name K. In the new line 40, the command assumes that K is a number that ranges from one upwards, and it counts the line numbers that are entered between commas. If K is one, H brings you to the first number; if K is two, H brings you to the second, and so on. You must make sure that there are as many line numbers following GOTO as there are choices on the menu.

Fig. 3 is a development of this system; you don't even have to ENTER! By using the INKEY$ command, whatever number you hit will be assigned to K at once, and your program choice follows. INKEY$ needs a string variable, K$, so that the step K = VAL (K$), or the use of ASC(K$), is needed to convert to the number form, K. Because TRS-80 BASIC has the ELSE command, the conversion can go into the same line. Line 40 makes the error trapping routine more interesting. If the selection has been correctly made line 40 is ignored, but a faulty selection causes the words INCORRECT ENTRY to be flashed ten times. This routine also makes use of STRING$ again.

When using PRINT STRING$, remember a number of characters are printed in a row. The number is the first number specified in the brackets. The second number is the ASCII code number for the character we want to print. If you can't be bothered to look it up, you can write the character between quotes, like STRING$(25,'*'). In this example, 32 is the ASCII code for a space, so STRING$(15,32) simply replaces the words INCORRECT ENTRY by spaces, deleting the words. Line 50 has the ON K GOTO selection feature, and the line numbers which follow take the program to the routines which are specified on the menu.

You can use the same program for different games, because you only have to select a different set of data and instructions for each game.

Alternatives

We don't always want a full menu selection. Sometimes a choice of two is quite enough. There are two methods I use. One is the letter or number method illustrated in Fig.4. The choice is between two items, and you are invited to type any number for one or for the other. Once again, we don't use ENTER when you make a choice—though it does give you time for second thoughts, because we use INKEY$. Line 10 gives the instructions, and line 20 contains the usual INKEY$ instructions. In line 30 we take VAL(K$), which will be zero if K$ is a letter, and use that to decide whether we jump to line 100 (PROCEED) or line 40 (RETURN). It's simple and effective, but it can be phased by typing 0 as a number. A foolproof way makes use of the ASCII codes of letters and numbers, and is shown in Fig. 5.

In line 30 of this program, the first section, K = ASC(K$) finds the ASCII code for the character which has been selected. If this is a number, then its ASCII code is less than 58 and more than 48, and this is sorted out by the second section of line 30. If the character is a letter, its ASCII code is less than 91 (unless you hit SHIFT as well) and more than 64. This also causes a jump. If any other key has been pressed, line 40 registers a mistake and causes a return, after a short delay, to the choice in line 10.

We can go further with the use of INKEY$. Fig. 6 shows a routine requesting a YES or NO answer directly from the keyboard without using ENTER. It's a development of the YES/NO routine we used in Part 4, with a flashing error message, and a flashing asterisk (which I call a 'flashterisk') as a prompt. Just to add bells and whistles, there is a time limit feature—you must type YES or NO quickly to beat the asterisk, or your entry is ignored!

These routines, ranging from the simple to the full scale YES/NO are often needed in a program. It is tedious to enter them in each place where needed.

That brings us to subroutines.

"The menu should... give the user some idea of what the choice is..."

```
5 REM INTO THE 8Ø'S FIG 6.2
1Ø CLS:DEFINT K,N
2Ø PRINTCHR$(23)TAB(15)"MENU":PRINTTAB(15)"****":PRINTT
     AB(1)"1. TO ENTER NEW DATA":PRINTTAB(1)"2. TO READ
      EXISTING DATA":PRINTTAB(1)"3. TO ENTER DATA FROM
     TAPE":PRINTTAB(1)"4. TO TERMINATE PROGRAM":PRINT"P
     LEASE CHOOSE BY NUMBER"
3Ø INPUT K:IF K<1 OR K>4 THEN CLS:PRINT"MISTAKE - PLEAS
     E TRY AGAIN":FOR N=1TO5ØØ:NEXT:GOTO2Ø
4Ø IF K=1 THEN 3ØØ
5Ø IF K=2 THEN 7ØØ
6Ø IF K=3 THEN 1ØØØ
7Ø IF K=4 THEN CLS:END
3ØØ CLS:PRINT"DATA ENTRY STARTS HERE":STOP
7ØØ CLS:PRINT"DATA READ SECTION STARTS HERE":STOP
1ØØØ CLS:PRINT"TAPE ENTRY SECTION STARTS HERE":STOP
```

Figure 2

```
5 REM INTO THE 8Ø'S FIG 6.3
1Ø DEFINT K,N
2Ø CLS:PRINTCHR$(23)TAB(15)"MENU":PRINTTAB(15)"####":PR
     INTTAB(2)"1. GAME OF GENDER":PRINTTAB(2)"2. GAME O
     F GROUPS":PRINTTAB(2)"3. GAME OF YOUNG":PRINTTAB(2
     )"4. TERMINATE":PRINT:PRINTTAB(2)"PLEASE SELECT BY
      NUMBER"
3Ø K$=INKEY$:IF K$="" THEN 3Ø ELSE K=VAL(K$)
4Ø IF K<1 OR K>4 THEN CLS:FOR N=1TO1Ø:PRINT@473,"INCORR
     ECT ENTRY":FOR Z=1TO2Ø:NEXT Z:PRINT@473,STRING$(15
     ,32):FOR Z=1TO2Ø:NEXT Z:NEXT N:GOTO2Ø
5Ø ON K GOTO 3ØØ,7ØØ,1ØØØ,5ØØØ
6Ø END
3ØØ CLS:PRINTTAB(25)"GAME OFGENDER":STOP
7ØØ CLS:PRINTTAB(25)"GAME OF GROUPS":STOP
1ØØØ CLS:PRINTTAB(25)"GAME OF YOUNG":STOP
5ØØØ CLS:END
```

Figure 3

```
5 REM INTO THE 8Ø'S FIG 6.4
1Ø PRINT"HIT ANY LETTER TO PROCEED, ANY NUMBER TO RETUR
     N"
2Ø K$=INKEY$:IF K$="" THEN 2Ø
3Ø IF VAL(K$)=Ø THEN 1ØØ ELSE4Ø
4Ø CLS:PRINT"RETURN PROGRAM STARTS HERE":STOP
1ØØ CLS:PRINT"PROCEED PROGRAM STARTS HERE":STOP
```

Figure 4

```
MENU
1. KEYBOARD ENTRY.
2. CASSETTE ENTRY.
3. TERMINATE.

PLEASE CHOOSE ★
```

Figure 1

Subroutines

A subroutine is a short (or long or middling but usually short) piece of program which is needed more than once in the course of a main program. It can be called up from different parts of the main program. Calling a subroutine means leaving your main program action and starting the subroutine action. It is implemented by the command GOSUB.

This is another very powerful command, because it saves having to type the same piece of program again and again. To see how it works, look at GOSUB in action in Fig. 7. Line 10 asks you to type any letter, and line 20 calls up the subroutine in line 100. This consists of the INKEY$ routine. The computer will wait for you to press a key. When that happens, the subroutine returns to the instruction after the place where it was called. In this case, that's the PRINT K$ instruction in line 20. The word LETTER is printed alongside. In line 30, you are asked to type any number, and once again the subroutine is called in line 40. This time the RETURN instruction in line 100 causes the number to be printed with NUMBER alongside because the return is in

"A subroutine is a piece of program which is needed more than once in the course of a main program."

```
5 REM INTO THE 80'S FIG 6.5
10 CLS:PRINT"HIT ANY LETTER TO PROCEED, ANY NUMBER TO R
     ETURN"
20 K$=INKEY$:IF K$="" THEN 20
30 K=ASC(K$):IF K>48 AND K<58 THEN 50 ELSE IF K>64 AND
     K<91 THEN 100
40 CLS:PRINT@480, "MISTAKE":FOR N=1TO500:NEXT:GOTO10
50 PRINT "RETURN PROGRAM":STOP
100 PRINT "PROCEED PROGRAM":STOP
```

Figure 5

```
999 REM INTO THE 80'S FIG 6.6
1000 CLS:A$=""
1010 K$=INKEY$:IF K$=""THEN 1200 ELSE PRINT K$;
1020 A$=A$+K$:IF LEN(A$)<2 THEN 1010
1030 IF LEN(A$)=2 AND A$="NO" THEN M=2:GOTO2000
1040 IF LEN(A$)=3 AND A$="YES" THEN M=1:GOTO2000
1050 IF LEN(A$)=2 GOTO1010 ELSE F$="MISTAKE":GOTO1500
1060 END
1200 PRINT@1,"*":FOR Z=1TO30:NEXT Z:PRINT@1," ":FOR Z=1
     TO30:NEXT Z:GOTO1010
1500 CLS:PRINTCHR$(23):FOR I=1TO15:PRINT@470,F$:FOR J=1
     TO20:NEXT J:PRINT@470,STRING$(20,32):FOR J=1TO20:N
     EXT J:NEXT I:PRINTCHR$(28):GOTO1000
2000 IF M=1 THEN CLS:PRINT"THE 'YES' PROGRAM FOLLOWS":E
     LSE CLS:PRINT "THE 'NO' PROGRAM FOLLOWS"
```

Figure 6

line 40. Just to be sure, we do it all over again in lines 50 and 60.

See the devilish cunning of it all? It's the same subroutine each time, but it's entered from different parts of the program. It returns to the instruction immediately following the GOSUB which called it.

You can have as many GOSUBs as you like, providing each one starts with a line number. You can't call up a subroutine which starts halfway along a line and ends with RETURN.

If you forget the RETURN, the program will crash through, going to the instruction which follows the last line of the subroutine. If there isn't one, the program will end, leaving you wondering what's happened.

If you enter a subroutine incorrectly, for example, forgetting the END in line 70 of Fig. 7, you'll get an error message in line 100—RG. This means RETURN without GOSUB because there is a return command, but no GOSUB to call it. There's no record inside the computer of where it should return. It can't return!

You can have a subroutine called from inside another subroutine. This is called nesting, and you can nest subroutines until you run out of memory.

Fig. 8 shows an example of a nested subroutine. The main program asks for a YES/NO answer, and this in turn causes a GOSUB to the INKEY$ routine we looked at earlier. This time, however, an error in the typing of YES or NO causes another subroutine to be called, a flashing error subroutine. Because this subroutine can be called from any part of the program, it is available to signal an error later on.

Use subroutines every time a piece of programming is done more than once in a program. The use of INKEY$ is one example. Another is any PRINT routine which is more than a simple PRINT N$ type of command.

A problem that turns up eventually when you start using subroutines is called passing parameters.

Look at the simple subroutine in Fig. 9. It compares two numbers, A and B, to determind which one is larger. This is perfectly straightforward if you have two numbers in the program which are represented by variables A and B. What happens if you haven't, or if you want to compare several sets? This is the problem of passing parameters. Whatever you want to compare has to be converted to the variable numbers A and B, because these are the variables which are used in the subroutine.

In line 10, there is no problem. The numbers are entered directly from the keyboard, and the comparison is made in line 20 by calling the subroutine. In line 30, two words are input, and we compare their lengths by making the variables A and B take the values of the word lengths. Then we call the subroutine. In line 50, two letters are input and their ASCII codes equated to the variables A and B so the subroutine can be used again to determine the order of the letters. An END command is used just BEFORE the subroutine to make certain that the subroutine cannot be entered accidentally, but must be called each time it is to be used.

Sometimes parameters have to be passed twice, once when the subroutine is being entered, and again after returning from it. Fig. 10 illustrates this, using a comparison of numbers which are the tag numbers of strings (the subscripts). In this simple example there is no reason why we should have used L$(N) in the subroutine. If you remember, however, that a subroutine like this would have to be called from several parts of a program—perhaps to sort out string variables—you see it is important to keep the variables used in the subroutine different from those in the main program. Unless we return to the original variables L$(N), the printout in line 40 will be incorrect, just a printout of the original strings.

If you have a subroutine which isn't working correctly, it could be that you're not passing parameters!

There's one more useful command which makes use of subroutines. It's a menu command, ON N GOSUB, and it works just like ON N GOTO. Your menu will list choices from one upwards, and ask for a choice which is then assigned to the variable SN. When the instruction ON N GOSUB is used, the program will branch to one of a number of subroutines. For example, if there were five menu items, we would need steps such as:

```
100 INPUT N: ON N GOSUB 200,300,400,500,600
```

Input 1, and you go to the subroutine which starts at line 200; input 2 and you go to the subroutine which starts a line 300, and so on. If you want less effort, you can use the INKEY$ answer instead of INPUT. Each subroutine will return to the instruction which follows the ON N GOSUB command, even if the next instruction is on the same line 100.

Neat printing

Messy printing is something that bugs

you once you get over the initial thrill of seeing a program work.

Professional progams are notable for good, clear, well set out print routines, and there's no reason why yours should look scruffy, especially when you can put your neat effort into a subroutine which can be used more than once, and in different programs. The main items needing attention are headings and underlining, boxing, and tabulation.

Headings are comparatively easy. The main thing is not to overkill. At the start of a program, it's sensible to have the title displayed in double sized letters, centered, with underlining, as illustrated in Fig. 12. If you have another 20 headings the same way though, it will tire the eye. Try grading your headings in order of importance, with double sized letters used once, apart from a flashing error warning.

The next important headings can use double-spaced letters with underlines, such as in line 30 of Fig. 11. The least important headings can be inset (using TAB(10)) and not underlined, but with a one line gap underneath.

There is no reason why you should follow that scheme, but it does illustrate what I mean. To match with the headings, print menus are in the same style.

To avoid looking at a set of instructions each time you run a program, contain the instructions in a subroutine. They can then be consulted at the start of a program if needed, but skipped if not.

A further refinement—if your program demands a lot of memory space, delete the instruction lines automatically if they are not needed. The DELETE command will run just as efficiently as a program instruction, as it does in direct command mode. Fig. 12 shows an example of this.

Boxing is another way to draw attention to something. This can be effective when a question is asked and an answer has to be typed—look at Fig. 13, and run it to see what happens.

The box is drawn in lines 20 and 30. In line 20 we pick the X-values of the ends of the box, and draw lines down, making the box three print lines deep. In line 30 we draw them across to complete the box. To program these effects, you need to use the video map on page E/1 of your manual. I clip a piece of tracing paper over the video map and draw the shapes I want on top of the paper. I can then see what has to be SET, to make the shape. It's easy if you want only straight horizontal or vertical lines. You can then use one FOR. . . . NEXT loop for the Y's and another for the X's.

Boxing can create some interesting effects. One is illustrated by adding the new line 50 shown in Fig. 14. Each character of

". . . the TRS-80 has a whole lot of tricks up its sleeve. One of the tricks. . . is the command ON K GOTO."

```
5 REM INTO THE 80'S FIG 6.7
10 CLS:PRINT "TYPE ANY LETTER"
20 GOSUB 100:PRINT K$;"  (LETTER)"
30 PRINT:PRINT"TYPE ANY NUMBER"
40 GOSUB 100:PRINT K$;"  (NUMBER)"
50 PRINT:PRINT"NOW TRY ANY KEY"
60 GOSUB 100:PRINT "YOU CONFUSED ME"
70 END
100 K$=INKEY$:IF K$="" THEN 100 ELSE RETURN
110 END
```

Figure 7

```
5 REM INTO THE 80'S FIG 6.8
10 CLS:PRINT"PLEASE TYPE YES OR NO (DON'T USE ENTER)":G
    OSUB 1000:PRINT :CLS:PRINT"YOUR CHOICE WAS ";M
20 END
1000 A$=""
1010 K$=INKEY$:IF K$="" THEN 1010 ELSE PRINT K$;
1020 A$=A$+K$:IF LEN(A$)<2 THEN 1010
1030 IF LEN(A$)=2 AND A$="NO" THEN M=2:RETURN
1040 IF LEN(A$)=3 AND A$="YES" THEN M=1:RETURN
1050 IF LEN(A$)=2 THEN 1010 ELSE F$="MISTAKE":GOSUB 120
    0:GOTO10
1200 CLS:PRINTCHR$(23):FOR I=1TO15:PRINT@470,F$:FOR J=1
    TO20:NEXT J:PRINT@470,STRING$(20,32):FOR J=1TO20:N
    EXT J:NEXT I:PRINTCHR$(28):RETURN
```

Figure 8

```
5 REM INTO THE 80'S FIG 6.9
10 CLS:INPUT "TWO NUMBERS,PLEASE";A,B
20 GOSUB510
30 INPUT"TWO WORDS, PLEASE"; N$,L$
40 A=LEN(N$):B=LEN(L$):GOSUB 510
50 INPUT "TWO LETTERS, PLEASE";A$,B$
60 A=ASC(A$):B=ASC(B$):GOSUB 510
70 END
500 END
510 IF A>B THEN CLS:PRINT "FIRST IS LARGER"
520 IF A=B THEN CLS:PRINT "THEY ARE EQUAL"
530 IF A<B THEN CLS:PRINT "SECOND IS LARGER"
540 RETURN
```

Figure 9

```
5 REM INTO THE 80'S FIG 6.10
10 REM YOU WOULD PLACE A DIM STATEMENT HERE
20 FOR N=1TO6:READ X(N),L$(N):NEXT
30 FOR N=1TO6 STEP2:A=X(N):B=X(N+1):Y$(N)=L$(N):Y$(N+1)
    =L$(N+1):GOSUB 200:L$(N)=Y$(N):L$(N+1)=Y$(N+1)
40 PRINT X(N);;;Y$(N);TAB(30)X(N+1);;Y$(N+1):NEXT
190 END
200 IF A>B THEN Z$=Y$(N):Y$(N)=Y$(N+1):Y$(N+1)=Z$
210 IF A=B THEN PRINT "EQUAL";
220 RETURN
400 DATA 2,"THUMB",1,"FINGER",7,"TOE",14,"FOOT",8,"JAW"
    ,8,"EAR"
410 END
```

Figure 10

```
5 REM INTO THE 80'S FIG 6.11
10 CLS:PRINT@344,CHR$(23)"HEADING":PRINTTAB(12)STRING$(
     7,48)
20 FOR N=1TO1200:NEXT:PRINT CHR$(28):PRINT TAB(13);
30 L$="SUBHEADING":FOR N=1TO LEN(L$):PRINT MID$(L$,N,1)
     ;"    ";:NEXT:PRINT:PRINTTAB(13)STRING$(37,48)
40 PRINT:PRINTTAB(6)"SUB-HEADING!":PRINT:PRINTTAB(2)"TH
     IS GIVES A REASONABLY NEAT APPEARANCE"
```

Figure 11

```
5 REM INTO THE 80'S FIG 6.12
10 CLS:PRINT@3,"DO YOU NEED INSTRUCTIONS?":GOSUB 1000:I
     F M=1 THEN GOSUB 5000 ELSE IF M=2 THEN DELETE 5000
     -5020
20 PRINT "NEXT STEP"
30 STOP
1000 REM THE YES/NO SUBROUTINE GOES HERE
1010 RETURN
5000 PRINTTAB(26)"INSTRUCTIONS":PRINTTAB(26)STRING$(12,
     48):PRINT
5010 PRINTTAB(2) "THE OPERATING INSTRUCTIONS GO HERE"
5020 RETURN
```

Figure 12

```
5 REM INTO THE 80'S FIG 6.13
10 CLS:PRINT@325,"WHAT IS YOUR NAME?"
20 FOR Y=18 TO 26:SET(32,Y):SET(92,Y):NEXT
30 FOR X=32 TO 92:SET(X,18):SET(X,26):NEXT
40 PRINT@465,"";:INPUT N$
50 PRINT@710,"YOUR NAME IS ";N$;" ,HUH?"
60 END
```

Figure 13

the name is peeled off by using the MID$ instruction, and at the same time part of the floor of the box is reset. The effect is of letters dropping down and knocking holes in the box, and it adds a bit of interest to what might be only a dull INPUT. Remember though, once per program is enough for these tricks.

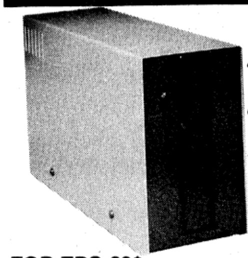
Tabulation is one of the things that can make a video screen or paper printout look really professional. You may not feel your programs need neat tabulation, but who knows? Take a look at Fig. 15, which uses string tabulations to round off the game from Part 3. This is an easy one because four columns can be set by using the comma as a delimiter. We then use a FOR-NEXT loop to print the items out, again using the commas as delimiters.

Another way of creating neat tabulation is to make use of the TAB instruction. It has been used in Fig. 16 to create a neat display of 90 random numbers. Line 10 sets up an array of 90 random numbers of two digits. The print tabulation is in line 20, using two FOR. . . NEXT loops. The first, FOR X = 1 to 90 STEP 9 sets up ten lines of numbers, and FOR Y = 0 to 8 creates the nine number positions across each line.

Fig. 17 shows the part of the routine which is of interest to us. Since this is a money table, we assume that the quantities are dollars and cents, and there are two figures after each decimal point. That means that if the last cents figures are lined up, the decimal points must also be lined up, and we can easily line up the right-hand side by using TAB and LEN. The figures are entered in line 10, and the variable used for the total. T is set to zero. In line 30 we set up another FOR-NEXT loop in which we calculate the total (T = T + Z(N)) and convert the quantity Z(N) to a string so that we can use LEN on it. In line 40, we print the value Z(N) at the tab position LEN(Z$), which starts before TAB(30). This spacing should be just right to get the end figures of the quantity on the TAB(30) position.

What do you do if someone enters a number not having the correct number of figures after the decimal point? The obvious answer (to me, anyhow) is to pack the number

WE WILL NOT BE UNDERSOLD



DISK DRIVES $314

40 track, 102K Bytes. Includes power supply and TRS-80* compatible silver enclosure. Ready to plug-in and run the moment you receive it. Can be intermixed with each other and Radio Shack drive on same cable. 90 day warranty. One year on power supply. Available for 220 Vac (50 Hz) operation. **External card edge included.**

FOR TRS-80*		
CCI-100	5¼", 40 Track (102K Bytes) for Model I	**$314**
CCI-280	5¼", 80 Track (204K Bytes) for Model I	**$429**
For Zenith Z89		
CCI-189	5¼", 40 Track (102K Bytes) add-on drive	**$394**
Z-87	Dual 5¼" add-on drive system	**$995**

DISKETTES — Box of 10 with plastic library case

5¼"	**Scotch $35**	**Maxell $40**	**BASF/Verbatim $24**
8"	**Scotch $50**	**Maxell $55**	**BASF/Verbatim $36**

CLEAR PLASTIC CASE-Holds 50 diskettes **$19**

NEW • S-100 CCS CARDS

MAINFRAME, Z-80 CPU, CONTROLLER, RAM, and 2P + 2S CARDS **$ CALL**

8" SHUGART SA801R DISK DRIVES $425

DISK OPERATING SYSTEMS

PATCHPAK #4 by Percom Data	**$ 8.95**
CP/M® for Model I, Zenith **$145** • for Model II, Altos	**$169.00**
NEWDOS Plus 40track	**$ 79.00**
NEWDOS 80	**$135.00**

COMPLETE SYSTEMS

ALTOS ACS8000 Computers	**$ CALL**
APPLE 16K	**$939**
APPLE III 96K	**$2999**
TRS-80* Model II-64K	**$3499**
TRS-80* Model III-16K	**$899**
TRS-80* Expansion interface	**$249**
ZENITH Z89, 48K all-in-one computer	**$2395**
ZENITH Z19	**$735**
TELEVIDEO 920C	**$748**
ATARI 400 $479 — **ATARI 800**	**$769**
APF Game Only $95 — **Complete System**	**$489**
MATTEL INTELLIVISION	**$229**

MONITORS

LEEDEX	12" B & W Video 100	**$129**
ZENITH	13" Color	**$379**
SANYO	9" B & W VM4509	**$145**
SANYO	12" B & W DM5012	**$210**
SANYO	12" Green Screen DM5112	**$215**
SANYO	13" Color DMC6013	**$375**
APF	9" B & W TVM-10	**$120**

TELECOMMUNICATIONS

LIVERMORE STAR MODEM 2-year guarantee	**$145**
UNIVERSAL DATA SYSTEMS UDS-103	**$179**
D-CAT HARD WIRED DIRECT MODEM	**$189**
AUTO-CAT Auto Answer, Direct Connect Modem	**$229**

COMMUNICATIONS SOFTWARE

CCI-TELNET VERSION 5: A communication package which enables microcomputer users to communicate both with large mainframes and other microcomputers. Completely CP/M compatible. Multiple communication protocols supported. **$149**

16K MEMORY UPGRADE KITS 2 for $65 $35

200 ns for TRS-80*, Apple II, (specify): **Jumpers $2.50**

PRINTERS

NEC Spinwriter
Letter Quality High Speed Printer
Includes TRS-80* interface software, quick change print fonts, 55 cps, bidirectional, high resolution plotting, graphing, proportional spacing: R.O. **$2395**
R.O. with Tractor Feed **$2595** KSR with Tractor Feed **$2795**

C.ITOH Starwriter, 25 CPS, daisy wheel printer	**$1795**
C.ITOH Starwriter II, 45 CPS. daisy wheel printer	**$1995**

Letter quality printers. Use up to 15" paper. 1 year warranty on parts. 3 months on labor. Proportional spacing and bidirectional printing. Same as VISTA V300.

EPSON MX-80	**$595**
PAPER TIGER **IDS 445** Graphics and 2K buffer	**$699**
IDS 460 Bidirectional, 160 cps, graphics and 2K buffer	**$1050**
IDS 560 132 Columns, graphics	**$1599**
ANADEX **DP-9500/01** **$1345** — **DP-8000**	**$849**
OKIDATA Microline 80 Friction and pin feed	**$525**
Tractor Feed, friction, and pin feed	**$625**
Microline 82 Bidirectional, friction and pin feed	**$719**
Microline 83 Bidirectional, 120 cps, uses up to 15" paper	**$995**
779 CENTRONICS TRACTOR FEED PRINTER Same as Radio Shack line printer I	**$969**
737 CENTRONICS FRICTION & PIN FEED PRINTER n × 9 proportional and 7 × 8 mono spacing. Same as Radio Shack line printer IV	**$780**
730 CENTRONICS FRICTION & PIN FEED PRINTER 7 × 7 matrix Same as Radio Shack line printer II	**$595**
P1 CENTRONICS PRINTER Same as Radio Shack quick printer	**$269**
EATON LRC 7000 + 64 columns, plain paper	**$269**
TI-810 Faster than Radio Shack line printer III. Parallel and serial w/TRS-80* interface software w/u&l case & paper tray	**$1589**
Compressed print, vertical form control	**$1865**

ACCESSORIES

SCOTCH HEAD CLEANING DISKETTE: Cleans drive Read/Write head in 30 seconds; specify 5¼" or 8".	**$ 25.00**
FLOPPY SAVER: Protection for center holes of 5¼" floppy disks. Installation tools and rings for 25 diskettes.	**$ 11.95**
Re-orders of rings only	**$ 6.95**
EXTERNAL DATA SEPARATOR: Eliminates data separation problems (crc). Improves reliability. This plug in unit comes fully assembled and tested.	**$ 29.95**
Z-80 SOFTCARD: Your key to software expansion. The plug-in Z-80 Softcard transforms your Apple into a Z-80 while keeping all the benefits of the 6502. Comes with CP/M in two disk format, MBASIC and GBASIC, full documentation and utility programs.	**$339.00**
VIDEX BOARD 80 Column, U/L case conversion card	**$299.00**
CRT FILM: Helps eliminate external glare, 9"	**$ 29.00**
RF MODULATOR: Adapts video to TV	**$ 29.00**
TRS-80 & OTHER MYSTERIES	**$ 18.95**
NEC SPINWRITER THIMBLE $11.95 — **RIBBON**	**$ 6.00**
CCS CARDS: Parallel or serial printer interface cards	**$115.00**
RS232: For Radio Shack Interface.	**$ 84.00**
DISK-DRIVE EXTENDER CABLES: Fits all mini-disk drives.	**$ 16.95**
SIX (6) PRONG ISOLATOR: ISO-2	**$ 54.00**
AC FILTER/6 PRONG POWER STRIP	**$ 39.00**
DISK DRIVE CABLES: 2 drive **$29.00** — 4 drive	**$ 35.00**
DUST COVERS: TRS-80/Apple	**$ 7.95**
PLASTIC DISKETTE HOLDER: For ring binder, holds 20	**$ 8.00**

For fast delivery, send certified checks, money orders or call to arrange direct bank wire transfers. Personal or company checks require two to three weeks to clear. All prices are mail order only.

DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED **Send for FREE Catalogue**

The CPU SHOP ✓298

TO ORDER CALL TOLL FREE 1-800-343-6522
TWX: 710-348-1796 Massachusetts Residents call 617/242-3361

5 Dexter Row, Dept. K02M
Charlestown, Massachusetts 02129
Hours 10AM-6PM (EST) Mon.-Fri. (Sat. till 5)

Technical Information call 617/242-3361
Massachusetts Residents add 5% Sales Tax
Tandy Corporation Trademark/® Digital Research

master charge THE INTERBANK CARD — VISA®

```
21 REM INTO THE 80'S FIG 6.14
10 CLS:PRINT@325,"WHAT IS YOUR NAME?"
20 FOR Y=18 TO 26:SET(32,Y):SET(92,Y):NEXT
30 FOR X=32 TO 92:SET(X,18):SET(X,26):NEXT
40 PRINT@465,"";:INPUT N$:FOR X=32 TO 92:SET(X,24):NEXT
50 FOR L=1 TO LEN(N$):RESET (36+2*L,24):PRINT@657+L,MID
    $(N$,L,1):FOR N=1TO 150:NEXT N:NEXT L
60 END
```

Figure 14

```
5 REM INTO THE 80'S FIG 6.15
10 FOR N=1TO4 :READ A$(N),F$(N),Y$(N),G$(N):NEXT
20 CLS:PRINT "MALE","FEMALE","YOUNG","GROUP":PRINT:PRIN
    T
30 FOR N=1TO4:PRINT A$(N),F$(N),Y$(N),G$(N):NEXT
40 DATA "GANDER","GOOSE","GOSLING","GAGGLE","BULL","COW
    ","CALF","HERD","RAM","EWE","LAMB","FLOCK","DOG","
    BITCH","PUPPY","PACK"
```

Figure 15

```
5 REM INTO`THE 80'S FIG 6.16
10 DIM N(100):FOR L=1TO90:N(L)=RND(99):NEXT
20 CLS:FOR X=1TO90 STEP 9:FOR Y=0TO8:PRINTTAB(Y*6+4)N(X
    +Y);:NEXT Y:PRINT:NEXT X
```

Figure 16

```
5 REM INTO THE 80'S FIG 6.17
10 CLEAR200:FOR N=1 TO 8:INPUT "CASH AMOUNT"; Z(N):NEXT
    :T=0
20 PRINT "CASH SUMS"
30 FOR N=1TO8:T=T+Z(N):Z$(N)=STR$(Z(N))
40 PRINTTAB(30 - LEN(Z$(N)))Z$(N):NEXT
50 PRINT:PRINTTAB(7)"TOTAL IS :-"TAB(30 - LEN(STR$(T)))
    T
```

Figure 17

with zeros until it has two figures after the point. The question is—how?

Fig. 18 will do just that. We have a new entry procedure here, which turns each number into a string Z$, and then tests Z$ to find where it has a decimal point. This is done by finding the length of Z$ and examining each character in turn, using the FOR-NEXT loop to see if a decimal point is present—ASCII code 46. If the figure, converted into a string, has two digits after the decimal point, it will be detected in line 30. If Z$ = '142.64' we will jump out of the loop in line 20 at K = 4, because the decimal point is the fourth character along from the start. The total number of characters is 6, so Z—K = 2, and we can jump to line 60 to print the amount.

Line 40 detects a figure with one digit after the decimal point. When this happens, Z – K = 1, do we can add a zero to pad the number string out. Finally, line 50 sorts out the last possibilities. If the number has been written with a decimal point but nothing after it, a pair of zeroes will be added, and if there is no decimal point (so that K will have taken a value of L + 1 before stopping the loop, and Z – K = – 1), then a decimal point and two zeroes are added.

This program applies to money quantities, but the techniques can be adapted for anything else where you need to recognize a feature and line up on it.

Tape on Tap

In Part 2 we looked at the CSAVE and CLOAD procedures for recording and replaying programs, to avoid the tedious task of having to key in a program each time you switch it on. You've probably discovered other chunks of information that you don't want to have to enter each time. You have a

✓Reader Service—see page 242

"When you CSAVE a program, the listing is saved. You don't have to do anything special to ensure this."

home finance program which you use once a week, you certainly don't want to spend the last week of the year reentering all the data for 51 previous weeks. You don't want to keep the computer running all year, so you can enter financial data once a week. You need to record the data once a week so that it can be recalled.

Some programs need more data than the computer can hold, though it may not be needed at once. In these cases, data has to be stored on cassette or some other storage system. A pack of recorded data is called a file, and data filing is a topic that gets more important and interesting. Cassette data files are called serial—you start recording at one end of the tape and you keep on until you're through or you hit the other end. There's no way you can automatically pick a piece out of the middle of the taped data without reading everything that's gone before, unless you note the tape counter readings of recorded sections. This problem drives most people to use disks because a disk system comes with an operating program (the Disk Operating System, DOS). It does the file-finding for you. Disk filing isn't all sweetness and light though. It's my opinion that most nonprofessional users don't need disk, especially now that there is an alternative, the Exatron Stringy-Floppy.

Back to cassette files. There's a record command and a replay command. The record command is PRINT#−1, and the replay command is INPUT#−1. When you play back, two asterisks appear, but they don't blink. The #−1 means that we have only one cassette recorder on line. If you have an expansion interface (which disqualifies you as a beginner), you can run two cassettes, #−1, and #−2. Since most people buy expansion interfaces to avoid cassettes, we'll stick to the #−1 channel, however, which uses the normal five-pin cassette connector at the back of the TRS-80.

There's a world of difference between the CSAVE and CLOAD commands and the PRINT#−1, and INPUT#−1 commands. When you CSAVE a program, the listing is saved. You don't have to do anything special to ensure this. Similarly, when you use CLOAD, you load in the whole program. PRINT#−1 and INPUT#−1 are different. You have to say if you are recording or replaying a string or a number, and the size is then restricted. The maximum safe size for one record operation is 248 characters; you can send out 255, but you can only get back 248.

In addition, you have to say what you are recording, and *no commas must appear, even within quotes*, in the string which is recorded.

Suppose you have two strings, L$ and S$, and two numbers, N and J, which you want to record. Your recording command will look something like:

100 PRINT# − 1, L$,S$,N,J

When that instruction comes along, you must be prepared with a cassette ready to record, and the record/play keys pressed. Usually we have a 'hit any key to start' step just before the recording stage, as shown in Fig. 19. The PRINT#−1 instruction in line 40 will record these items on the tape, along with a leader and a brief trailer (end byte). Each PRINT#−1 command causes the leader to be recorded, followed by the data, then the trailer, so that quite a lot of tape will be used even if there is only one byte of data to be recorded.

To replay the data, you need a section which contains an INPUT#−1 command with the same arrangement of strings and numbers as the PRINT#−1 command. The variable names don't have to be the same, but the order and number of the variable data must be. If we want to replay the data recorded by the PRINT −1 command used in the previous example, we could use an instruction such as:

INPUT# − 1, A$.B$,C,D

This uses different variable names, but the arrangement is identical—two strings followed by two numbers. Any other order, or a

```
5 REM INTo THE 80'S FIG 6.18
10 CLEAR 200:FOR N=1TO8:INPUT "CASH AMOUNT";Z$(N):Z=LEN
    (Z$(N))
20 FOR K=1 TO Z:IF MID$(Z$(N),K,1)<>"." THEN NEXT K
30 IF Z-K=2 THEN 55
40 IF Z-K =1 THEN Z$(N)=Z$(N)+"0":GOTO55
50 IF Z-K=0 THEN Z$(N)=Z$(N)+"00" ELSE Z$(N)=Z$(N)+".00
"
55 NEXT N
60 T=0:CLS:PRINT"CASH SUMS"
70 FOR N=1TO8:T=T+VAL(Z$(N))
80 PRINTTAB(30-LEN(Z$(N)))Z$(N):NEXT
90 PRINT:PRINTTAB(7)"TOTAL IS :- "TAB(30-LEN(STR$(T)))T
100 END
```

Figure 18

```
5 REM INTO THE 80'S FIG 6.19
10 INPUT "TWO WORDS, PLEASE";L$,S$
20 INPUT"...AND NOW TWO NUMBERS";N,J
30 CLS:PRINT@135,"PREPARE FOR RECORDING DATA,PLEASE":PR
    INT:PRINT"PRESS ANY KEY TO START RECORDER"
40 K$=INKEY$:IF K$="" THEN 40
50 PRINT#-1,L$,S$,N,J
60 CLS:PRINT "RECORDING COMPLETE - PRESS STOP KEY ON RE
    CORDER"
70 END
```

Figure 19

```
5 REM INTO THE 80'S FIG 6.20
10 CLEAR300:S$="":FOR J=1TO50
20 INPUT "ENTER A FOUR-FIGURE NUMBER,PLEASE";N:N$=RIGHT
    $(STR$(N),4)
30 S$=S$+N$:NEXT
40 CLS:PRINT@330,"PLEASE PREPARE FOR RECORDING - PRESS
    ANY KEY TO START"
50 K$=INKEY$:IF K$="" THEN 50
60 PRINT#-1,S$
70 CLS:PRINT"RECORDING COMPLETE":PRINT "S$ IS ";S$
```

Figure 20

PROGRAMMING TOOLS FOR YOUR TRS-80

INSIDE LEVEL II

The Programmers Guide to the TRS-80 ROMS

INSIDE LEVEL II is a comprehensive reference guide to the Level II ROMs which allows the machine language or Basic programmer to easily utilize the sophisticated routines they contain. Concisely explains set-ups, calling sequences, and variable passage for number conversion, arithmetic operations, and mathematical functions, as well as keyboard, tape, and video routines. Part II presents an entirely new composite program structure which loads under the SYSTEM command and executes in both Basic and machine code with the speed and efficiency of a compiler. In addition, the 18 chapters include a large body of other information useful to the programmer including tape formats, RAM useage, relocation of Basic programs, USR call expansion, creating SYSTEM tapes of your own programs, interfacing of Basic variables directly with machine code, a method of greatly increasing the speed at which data elements are stored on tape, and special precautions for disk systems. INSIDE LEVEL II is a clearly organized reference manual. It is fully typeset and packed with nothing but useful information. It does not contain questions and answers, ROM dumps, or cartoons. **INSIDE LEVEL II.....$15.95**

4 SPEED OPTIONS FOR YOUR TRS-80!

The SK-2 clock modification allows CPU speeds to be switched between normal, an increase of 50%, or a 50% reduction; selectable at any time without interrupting execution or crashing the program. Instructions are also given for a 100% increase to 3.54 MHz, though the TRS-80 is not reliable at this speed. The SK-2 may be configured by the user to change speed with a toggle switch or on software command. It will automatically return to normal speed any time a disk is active, requires no change to the operating system, and has provisions for adding an LED to indicate when the computer is not at normal speed. It mounts inside the keyboard unit with only 4 necessary connections for the switch option (switch not included), and is easily removed if the computer ever needs service. The SK-2 comes fully assembled with socketed IC's and illustrated instructions. **SK-2.....$24.95**

PROGRAM INDEX FOR DISK BASIC

Assemble an alphabetized index of your entire program library from disk directories. Program names and free space are read automatically (need not be typed in) and may be alphabetized with a fast Shell/Metzner sort by disk or program. The list may also be searched for any disk, program, or extension; disks or programs added or deleted; and the whole list or any part sent to the printer. Finally, the list itself may be stored on disk for future access and update. "The best thing since sliced bread" (January issue of '80 Microcomputing). Works with TRSDOS, NEWDOS, and NEWDOS/80. One drive and 32K required. **INDEX.....$19.95**

RAM SPOOLER AND PRINT FORMATTER

This program is a full feature print formatting package featuring user defineable line and page length (with line feeds inserted between words or after punctuation), screen dump, printer pause control, and baud rate selection. In addition, printing is done from a 4K expandable buffer area so that the LPRINT or LLIST command returns control to the user while printing is being done. Ideal for Selectric or other slow printers. Allows printing and processing to run concurrently. Output may be directed to either the parallel port, serial port, or the video screen. **SPOOLER.....$16.95**

TELECOMMUNICATIONS PROGRAM

This machine language program allows reliable high speed file transfers between two disk-based computers over modems or direct wire. It is menu driven and extremely simple to use. Functions include real-time terminal mode, save RAM buffer on disk, transmit disk file, receive binary files, examine and modify UART parameters, program 8 custom log-on messages, automatic 16-bit checksum verification of accurate transmission and reception, and many more user conveniences. Supports line printers and lower-case characters. With this program you will no longer need to convert machine language programs to ASCII for transmission, and you will know immediately if the transmission was accurate. **TELCOM.....$29.95**

SINGLE STEP THROUGH RAM OR ROM

STEP80 allows you to step through any Basic or machine language program one instruction at a time, and see the address, hexadecimal value, Zilog mnemonic, register contents, and step count for each instruction. The top 14 lines of the video screen are left unaltered so that the "target program" may perform its display functions unobstructed. STEP80 will follow program flow right into the ROMs, and is an invaluable aid in learning how the ROM routines function. Commands include step (trace), disassemble, run in step mode at variable step rate, display or alter memory or CPU registers, jump to memory location, execute a CALL, set breakpoints in RAM or ROM, write SYSTEM tapes, and relocate to any page in RAM. The display may also be routed to your line printer through the device control block so custom print drivers are automatically supported. **STEP80.....$16.95**

MACHINE CODE FAST FOURIER TRANSFORM

This complete package includes 3 versions of the machine language FFTASM routine assembled for 16, 32, and 48K machines, a short sample Basic program to access them, a 10K Basic program which includes sophisticated interactive graphing and data manipulation, and a manual of instructions and examples. The machine language subroutines use variables defined by a supporting Basic program to make data entry and retrieval extremely fast and easy for custom implementation. They perform 20 to 40 times faster than their Basic equivalent (256 points in 12.5 seconds), and require less than 1550 bytes of memory. The FFT is useful in analyzing stock market and comodity trends as well as for scientific information. **FFTASM.....$49.95**

DUPLICATE SYSTEM TAPES WITH CLONE

Make duplicate copies of any tape written for Level II. They may be SYSTEM tapes (continuous or not) or data lists. The file name, load address, entry point, and every byte (in ASCII format) are displayed on the video screen. **CLONE.....$16.95**

EDIT BASIC PROGRAMS WITH ELECTRIC PENCIL

Load Basic programs or any other ASCII data file into the disk version of Electric Pencil for editting. One command from DOS quickly modifies existing files to Pencil format. One disk and 32K required. **PENPATCH.....$9.95**

RAMTEST FOR LEVEL II

This machine language program is a very thorough test for several types of RAM errors. A complete test of each individual bit in a 48K machine takes just 14 seconds. Includes a separate test for power line glitches. **RAMTEST.....$9.95**

MUMFORD MICRO SYSTEMS ✓144

ORDERING: Complete satisfaction is guaranteed or a full refund will be made. All Model I programs are shipped on cassette unless $5 is included for a formatted (no system) disk. Include $1 postage and handling. California residents add 6% sales tax. Visa, Master-charge and COD orders accepted.

Box 435-E Summerland, California 93067 (805) 969-4557

✓ Reader Service—see page 242

```
5 REM INTO THE 80'S FIG 6.21
500 DIM L(51):PRINT"PLEASE PREPARE FOR REPLAY. HIT ANY
     KEY WHEN READY":L$="":N=0
510 K$=INKEY$:IF K$="" THEN 510
520 INPUT#-1,L$
530 FOR N=1TO50:L(N)=VAL(MID$(L$,4*N-3,4)):NEXT
540 PRINT"DATA READY - ":FOR Z=1TO500:NEXT:FOR N=1TO50
     STEP 10:FOR X=0TO9:PRINT L(N+X);:NEXT X:PRINT:NEXT
     N
550 END
```

Figure 21

```
5 REM INTO THE 80'S FIG 6.22
10 S$="":FOR J=1 TO 30:PRINTJ".";:INPUT N
20 N$=STR$(N):N$=STRING$(7-LEN(N$),32)+N$:S$=S$+N$:NEXT
30 REM DATA IS NOW PACKED
40 PRINT:PRINT"PACKED DATA - ";S$
50 REM NOW RECORD IT!
```

Figure 22

```
5 REM INTO THE 80'S FIG 6.23
1000 T=0: REM DATA MUST BE IN SINGLE STRING FORM. THE M
     ESSAGE WILL ALREADY HAVE PRINTED THAT RECORDER IS
     TO BE MADE READY, ENTER PRESSED TO START RECORDING
1010 PRINT#-1, A$:IF T=1 THEN 1040
1020 PRINT"NOW PREPARE BACKUP CASSETTE":T=1
1030 CLS:PRINT"PRESS ENTER WHEN READY TO RECORD BACKUP"
     :INPUT X:GOTO1010
1040 CLS:PRINT "RECORDING COMPLETE":REM PROGRAM THEN PR
     OCEEDS
```

Figure 23

differing number of string or number variables, will cause an error message, FD (faulty data) if the sequence is wrong, or OD (out of data) if you have asked for more data in the INPUT command than was recorded. It's not a bad idea to use different variable names on the replay.

This is very straightforward stuff, but as it is (and the manual isn't helpful on this point), it requires time-consuming routines. It uses a lot of tape. If you set up a loop which looks something like:

```
200 for J = 1 to 100: PRINT# – 1,N(J):NEXT
```

you'll have time to take in a month's repeats of 'Dallas' while it records, because each step in the FOR-NEXT loop starts a new recording with leader and trailer. This line will cause 100 recording runs!

Pack it close

Since you probably bought a computer to save time, this lengthy procedure is useless. All would be well if we could just use a line like 300 PRINT – 1, FOR J = 1 to 100:N(J):NEXT—but we can't. The problem is to pack the data so that 240 bytes or so can be recorded in one chunk. I haven't seen this topic discussed very much in magazine articles, but it's one I've spent time on. It could be that this will help even hardened, old time operators.

The solution is simple, if the data consists of numbers or strings which are the same length. If they are numbers, convert them into strings, using the STR$(number) command. Remember, this packs strings into long strings, using the + (concatenate) string action, and records this long string.

Fig. 20 shows what has to be done. Line 10 sets up a FOR-NEXT loop to input 50 numbers of four digits each, and the DIM statement prepares for this. Each number is converted into a string as it is entered, and the length is tested to make sure you don't cheat. These numbers can come from any part of the program.

The packing routine is on line 40. S$ was initialized as a blank string in line 10, and it now has the number string tagged on. After three strings, for example 1234, 5678, and 9012, the string S$ is 123456789012. This

"The maximum safe size for one record operation is 248 characters; you can send out 255, but you can only get back 248."

```
5 REM&INTO(THE080'S499 STOP
500 CLS:PRINT"PREPARE A CASSETTE FOR A TAPE FILE"
510 PRINT"NOTE THE STARTING POINT ON THE TAPE COUNTER,
     AND PRESS THE PLAY AND RECORD KEYS"
520 PRINT"PRESS ENTER WHEN READY"
530 INPUT X:CLS:PRINTTAB(21)"RECORDING...PLEASE WAIT"
540 PRINT #-1,I: REM I IS THE NUMBER OF ITEMS
550 A$=""
560 FOR N=1 TO I:A$=A$+L$(N)+CHR$(128)
570 IF LEN(A$)+LEN(L$(N+1)<245 THEN 590
580 PRINT#-1,A$:A$=""
590 NEXT N:PRINT #-1,A$
600 CLS:PRINT" RECORDING FINISHED. PRESS ENTER TO RETUR
     N TO MENU":REM NEED A RETURN TO MENU ROUTINE HERE
610 STOP: REM REPLAY ROUTINE STARTS HERE
620 CLS:PRINT@336,"PREPARE THE DATA TAPE FOR REPLAY"
630 PRINTTAB(13)"PRESS PLAY KEY; WHEN READY PRESS ENTER
    "
640 INPUT X:CLS:PRINTTAB(19)"ENTERING DATA, PLEASE WAIT
     ":X=1
650 INPUT#-1,I
660 INPUT#-1,A$:FOR N=1TO245:B$=MID$(A$,N,1)
670 IF B$<>CHR$(128) THEN L$(X)=L$(X)+B$:GOTO690
680 X=X+1
690 NEXT N:IF X<I GOTO660
700 CLS:PRINTTAB(26)"DATA ENTERED."
710 REM NOW YOU DISPLAY, OR OTHERWISE USE DATA
```

Figure 24

string increases until all 50 numbers have been joined, and S$ is 200 characters. The long string is recorded. There will be a leader and a trailer recorded with it, and you have saved a lot of recording time.

If the number existed in the form of N(J) before the packing step, you will need a FOR-NEXT loop which converts each number into a string and then packs it. Make sure that S$ is set to blank (") before the FOR-NEXT loop which packs it. Otherwise you can get some peculiar results when you do the routine more than once.

Replaying a packed string is easy, provided you know how it was packed. In this example, we used data in four-digit units, 50 to a string. Our replay procedure looks something like Fig. 21. Lines 500–520 are the usual replaying procedure, rewound to the correct place, ready for replay. At 520, the 200-byte string which we've labelled L$ will read in from the cassette. Converting this into the form we need, 50 sets of numbers, is done in line 530. The FOR-NEXT loop sets the number as 50, and the expression L(N) = VAL(MID$(L$,4∗N – 3,4)) gets the groups.

When N = 1, we try to find VAL(MID$(L$, 4∗1 – 3,4) which is the value of the group of four characters starting with character 1. That's the first set of four. When N = 2, 4∗N – 3 is 5, we read another four starting with character five. That's the second set of four.

The key to this is the formula 4∗N – 3 which we've used to find the first character from 1 to 50. Whatever number of digits you use to a group, it is always similar; it's the number of coded digits multiplied by N, with one less than the group number subtracted; if you are dealing with seven-character groups, the formula would be 7∗N – 6.

This style of packing and unpacking can make cassette files more efficient than the Level II manual suggests. It can even delay abandoning cassettes!

Suppose that the items you record are not in convenient groups of four or whatever? One answer is to pack so that they are a standard length. Suppose the items are single-precision numbers with up to six digits. There's no reason why any number of less than six shouldn't be packed with blanks up to six-digit length, using something like Fig. 22. The key part of the routine is in line 20:

```
N$ = STRING$(7-LEN(N$),32) + N$
```

If N$ is 25.2, then the length of the string is five characters because the STB$ conversion always adds a leading blank. The instruction is to form a number of blanks (ASCII 32) equal to 7 – 5, and add these to N$, making N$ two blanks longer. When this lot is unpacked, the VAL command will simply remove these blanks again.

Maybe you're hard to please, and you want to pack together strings of different length. You'll object to packing your valuable tape with blanks, which can happen if some are one or two characters and others 20 or more. There are two solutions in BASIC which I use (and others in machine code). I'll describe the simplest of the BASIC methods here—it's rather slow, but it works well.

The slow routine depends on the use of ASCII character 128. This is a blank, like ASCII 32, but with a difference. ASCII is what you get when you hit space-bar on the keyboard, but ASCII 128 never gets entered from the keyboard, and the computer recognizes it as a different character. If we pack strings with 128 between them, we should be able to unpack them by scanning the replayed string and looking for the 128 code number. This is what makes the replay slow because all the replayed characters have to be checked. There is an enormous saving in time, compared with recording each string. The speed of the routine doesn't matter if the strings are displayed on the video screen. If anything, we'll probably want to slow things down.

Fig. 23 shows the routines. The packing is fairly straightforward, with the long string formed with a CHR$(128) between each added string. Make sure that the total number N is recorded to make playback easy. Another additon is the string length detecting routine in line 570. This ensures that the string does not become too long to record because you don't know how many you can pack. If one more string makes the total too long, it is recorded, then reset to zero so that packing can continue.

The de-packing routine examines each character of a replayed long string until a 128 appears. The assembled string is given a subscript number, the number is incremented, and the depack routine continues. The end comes when the subscript number equals the number of strings recorded, or when an end of data code is detected. I've opted for a recorded number in this example.

Routines like these convert cassette data files from rather useless curiosities into reasonable methods of storing and replaying data. The high-speed methods using machine code (with routines built into the TRS-80 ROM) can be impressive.

One warning—always make a backup cassette of valuable data just as you make a backup of a valuable program. It's worth while to put a special routine in your programs to do this, such as Fig. 24.

Next month—Planning programs, PEEKING, POKING, POINTING, and what to do now that you're no longer a beginner.■

Kick your TRSDOS clock back on time after re-boot.

Clock Boot

Yuergen Boehmke
1365 W. 69 St.
Hialeah, FL 33014

System software clocks are nice until there's a need to reboot the system and reset the clock by you know who. In order to escape this chore, I sat down one evening and devised a simple means to let the computer work for me and not me for the computer. I devised a simple yet effective machine language program to continually update a semi-protected area of memory as frequently as the keyboard is scanned.

The program is best loaded in high memory and should be user-protected against other programs. It was originally written to be run with my TRS-80 DOS, however, it functions on Level I or II with or without the expansion interface, if you write a simple clock program and provide for automatic loading upon reboot. (Similar to the AUTO feature of TRS-80 Disk Operating System).

Upon initial entry, the program is entered at symbolic location INIT, which is used to modify the keyboard scan routine address from 4371H to BC00H. This allows my program to be run just prior to keyboard scanning.

After the clock program completes its function, it returns control to the scanning routine with a JP 4371H instruction. Each time it is entered, a check is made as to whether the year = 0, thus signifying that a reboot has occurred.

If a reboot has not occurred, address location BBFAH is updated with the current date and time. If a reboot has indeed occurred, address location 4041H is updated from the last known correct date and time stored in BBFAH. 4041H is the starting address where the system stores the current date and time.

After reboot, the clock will lose several seconds due to the time required to reload the Disk Operating System. If this loss is crucial to your programs, a slight modification could add any lost time.

It is best to locate the temporary time location, in this case, BBFAH, just above the main program. Take this precaution through reserved memory, starting at temporary time location, BBFAH.

When you save the program on disk, specify symbolic location CLOCK as the start address, and symbolic location INIT as the entry point, or it just won't work, since the temporary storage location will be overwritten on every loading.■

```
;  AUTOMATIC SYSTEM CLOCK UPDATE ROUTINE
;  REQUIRMENTS: TRS-80 LEVEL II WITH EXPANSION INTERFACE OR
;               TRS-80 LEVEL I WITH USER WRITTEN SOFTWARE CLOCK PROGRAM
;  MEMORY REQUIREMENTS:  42 BYTES
;  WRITTEN BY:  YUERGEN BOEHMKE
;               1365 WEST 69 STREET
;               HIALEAH, FLORIDA  33014
;  ORIGINAL VERSION:  01/02/80
;
                        ORG   0BC00H
BC00 214440   CLOCK     LD    HL,4044H      ; SYSTEM STORAGE FOR YEAR
BC03 7E                 LD    A,(HL)
BC04 FE00               CP    00H           ; CHECK IF YEAR = 0
BC06 21FABB             LD    HL,0BBFAH     ; TEMP TIME STORAGE
BC09 114140             LD    DE,4041H      ; SYSTEM STORAGE FOR TIME
BC0C CA10BC             JP    Z,C1          ; JUMP IF REBOOT OCCURED
BC0F EB                 EX    DE,HL         ; EXCHANGE REGISTER IF TIME OK
BC10 0606     C1        LD    B,06H         ; CYCLE COUNT
BC12 7E       C2        LD    A,(HL)        ; UPDATE TIME AND DATE
BC13 12                 LD    (DE),A
BC14 23                 INC   HL
BC15 13                 INC   DE
BC16 10FA               DJNZ  C2            ; CONTINUE UNTIL UPDATED
BC18 C37143             JP    4371H         ; RETURN TO SCAN PROGRAM
BC1B 2100BC   INIT      LD    HL,0BC00H     ; MODIFY SCAN ROUTINE
BC1E 221640             LD    (4016H),HL
BC21 C300BC             JP    CLOCK         ; JUMP TO CLOCK ROUTINE
                        END   INIT

TOTAL ERRORS = 0000
```

Program Listing 1

✓Reader Service—see page 242

✓ Reader Service—see page 242

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

NEW

BUSINESS PAC 100

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software (less a $3 penalty for handling)

100 Ready-To-Run Business Programs

(ON CASSETTE OR DISKETTE).....Includes 110 Page Users Manual.....5 Cassettes (Or Diskettes)
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS 100 PROGRAM LIST

	NAME	DESCRIPTION
1	RULE78	Interest Apportionment by Rule of the 78's
2	ANNU1	Annuity computation program
3	DATE	Time between dates
4	DAYYEAR	Day of year a particular date falls on
5	LEASEINT	Interest rate on lease
6	BREAKEVN	Breakeven analysis
7	DEPRSL	Straightline depreciation
8	DEPRSY	Sum of the digits depreciation
9	DEPRDB	Declining balance depreciation
10	DEPRDDB	Double declining balance depreciation
11	TAXDEP	Cash flow vs. depreciation tables
12	CHECK2	Prints NEBS checks along with daily register
13	CHECKBK1	Checkbook maintenance program
14	MORTGAGE/A	Mortgage amortization table
15	MULTMON	Computes time needed for money to double, triple, etc.
16	SALVAGE	Determines salvage value of an investment
17	RRVARIN	Rate of return on investment with variable inflows
18	RRCONST	Rate of return on investment with constant inflows
19	EFFECT	Effective interest rate of a loan
20	FVAL	Future value of an investment (compound interest)
21	PVAL	Present value of a future amount
22	LOANPAY	Amount of payment on a loan
23	REGWITH	Equal withdrawals from investment to leave 0 over
24	SIMPDISK	Simple discount analysis
25	DATEVAL	Equivalent & nonequivalent dated values for oblig.
26	ANNUDEF	Present value of deferred annuities
27	MARKUP	% Markup analysis for items
28	SINKFUND	Sinking fund amortization program
29	BONDVAL	Value of a bond
30	DEPLETE	Depletion analysis
31	BLACKSH	Black Scholes options analysis
32	STOCVAL1	Expected return on stock via discounts dividends
33	WARVAL	Value of a warrant
34	BONDVAL2	Value of a bond
35	EPSEST	Estimate of future earnings per share for company
36	BETAALPH	Computes alpha and beta variables for stock
37	SHARPE1	Portfolio selection model-i.e. what stocks to hold
38	OPTWRITE	Option writing computations
39	RTVAL	Value of a right
40	EXPVAL	Expected value analysis
41	BAYES	Bayesian decisions
42	VALPRINF	Value of perfect information
43	VALADINF	Value of additional information
44	UTILITY	Derives utility function
45	SIMPLEX	Linear programming solution by simplex method
46	TRANS	Transportation method for linear programming
47	EOQ	Economic order quantity inventory model
48	QUEUE1	Single server queueing (waiting line) model
49	CVP	Cost-volume-profit analysis
50	CONDPROF	Conditional profit tables
51	OPTLOSS	Opportunity loss tables
52	FQUOQ	Fixed quantity economic order quantity model
53	FQEOWSH	As above but with shortages permitted
54	FQEOQPB	As above but with quantity price breaks
55	QUEUECB	Cost-benefit waiting line analysis
56	NCFANAL	Net cash-flow analysis for simple investment
57	PROFIND	Profitability index of a project
58	CAP1	Cap. Asset Pr. Model analysis of project
59	WACC	Weighted average cost of capital
60	COMPBAL	True rate on loan with compensating bal. required
61	DISCBAL	True rate on discounted loan
62	MERGANAL	Merger analysis computations
63	FINRAT	Financial ratios for a firm
64	NPV	Net present value of project
65	PRINDLAS	Laspeyres price index
66	PRINDPA	Paasche price index
67	SEASIND	Constructs seasonal quantity indices for company
68	TIMETR	Time series analysis linear trend
69	TIMEMOV	Time series analysis moving average trend
70	FUPRINF	Future price estimation with inflation
71	MAILPAC	Mailing list system
72	LETWRT	Letter writing system-links with MAILPAC
73	SORT3	Sorts list of names
74	LABEL1	Shipping label maker
75	LABEL2	Name label maker
76	BUSBUD	DOME business bookkeeping system
77	TIMECLCK	Computes weeks total hours from timeclock info.
78	ACCTPAY	In memory accounts payable system-storage permitted
79	INVOICE	Generate invoice on screen and print on printer
80	INVENT2	In memory inventory control system
81	TELDIR	Computerized telephone directory
82	TIMUSAN	Time use analysis
83	ASSIGN	Use of assignment algorithm for optimal job assign.
84	ACCTREC	In memory accounts receivable system-storage ok
85	TERMSPAY	Compares 3 methods of repayment of loans
86	PAYNET	Computes gross pay required for given net
87	SELLPR	Computes selling price for given after tax amount
88	ARBCOMP	Arbitrage computations
89	DEPRSF	Sinking fund depreciation
90	UPSZONE	Finds UPS zones from zip code
91	ENVELOPE	Types envelope including return address
92	AUTOEXP	Automobile expense analysis
93	INSFILE	Insurance policy file
94	PAYROLL2	In memory payroll system
95	DILANAL	Dilution analysis
96	LOANAFFD	Loan amount a borrower can afford
97	RENTPRCH	Purchase price for rental property
98	SALELEAS	Sale-leaseback analysis
99	RRCONVBD	Investor's rate of return on convertable bond
100	PORTVAL9	Stock market portfolio storage-valuation program

☐ **CASSETTE VERSION $ 99.95**
☐ **DISKETTE VERSION $ 99.95**
☐ **MODEL II VERSION $149.95**

ADD $2.00 FOR SHIPPING IN UPS AREAS
ADD $3.00 FOR C.O.D. OR NON-UPS AREAS
ADD $4.00 OUTSIDE U.S.A, CANADA & MEXICO

H & E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

24 HOUR ORDER LINE
(914) 425-1535

✓Reader Service—see page 242

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

SMALL BUSINESS SYSTEMS GROUP

COORDINATED BUSINESS SYSTEMS

★ *All orders processed within 24-Hours*
★ *30-Day money back guarantee on all Software*
★ *Add $2.00 for shipping in UPS Areas*
★ *Add $3.00 for C.O.D. or NON-UPS Areas*
★ *Add $4.00 outside U.S.A, Canada & Mexico*

FACTS ABOUT THE S.B.S.G. BUSINESS PACKAGES

1. **S.B.S.G.** is a sophisticated Business Software System designed for the serious businessman.
2. Each of the **S.B.S.G. Business Modules** may be purchased separately...or you may purchase the entire coordinated business system.
3. Modules purchased separately do not coordinate with the General Ledger (although for the standard **S.B.S.G.** fee, the user may upgrade his individual modules for the coordinated system).
4. Foolproof, Step-By-Step procedures are supplied, planned and documented for the **First-Time Computer User.** All programs are self-explanatory, telling the user what is required at every step.
5. Programs are written in **BASIC** and the source code listing is supplied for those users who decide to modify the original system.
6. A complete users manual is supplied with each module.
7. Demo Data diskettes are supplied with sample data.
8. **S.B.S.G.** has an In-House staff that can answer questions and problems related to the proper use of the **S.B.S.G. Business System** (on the telephone or through the mail).
9. First-Time Computer Owners Note-Instructions are provided for entering state payroll withholding tables. There is an additional charge if you prefer to have **S.B.S.G. Programmers** insert the correct data.
10. Minimum system requirement is 2-drives to run any single module.
11. Minimum system requirement is 3-drives to run the coordinated business system (AR-AP-GL) or (AR-AP-GL with PAYROLL).
12. Minimum system requirement is 4-drives to run the extended coordinated system (AR-AP-GL-PR and INVENTORY/INVOICING).
13. The **A. OSBORNE & ASSOCIATES** business manuals are provided **FREE** with each order (they may be purchased separately at $20 per manual).
14. The INVENTORY and INVOICING modules are original programs written by **S.B.S.G.**
15. Each module can be purchased as independent modules to run on a 2 or more drive system except INVOICING.
16. Memory requirement is 48K for the MODEL-I and 64K for the MODEL-II.
17. All **S.B.S.G. BUSINESS SYSTEMS** may be upgraded up to 4-disk drives. No data is ever lost during an upgrade. There is a standard **S.B.S.G.** charge for all upgrades.

ACCOUNTS PAYABLE

The accounts payable system receives data concerning purchases from suppliers and produces checks in payment of outstanding invoices. In addition, it produces cash management reports. This system aids in tight financial control over all cash disbursements of the business. Several reports are available and supply information needed for the analysis of payments, expenses, purchases and cash requirements. All A/P data feeds General Ledger so that data is entered into the system just once. These programs were developed 5 years ago for the Wang micro-computer and have been tested in many environments since then. The package has been converted to the TRS-80™ and is now well documented, on-line, interactive micro-computer system with the capabilities of (or exceeding many larger systems).

CAPABILITIES:

★ menu driven; easy to use; full screen prompting and cursor control
★ invoice oriented; everything revolves around the invoice; handles new invoice or credit memo or debit memo
★ invoce information recorded; invoice #, description, buyer, check register #, invoice date, age date, amount of invoice, discount (in %), freight, tax ($), total payable
★ transaction print and file maintenance procedures insure accuracy
★ flexible check calculation procedure; allows checks to be calculated for a set of vendors-or-for specific vendors
★ program prints your checks; contiguous computer checks with your company letterhead can be purchased from SBSG
★ reports include (samples on back):
- open item listing/closed item listing - both detail and summary
- debit memo listing/credit memo listing
- aging
- check register report (to give an audit trail of checks printed)
- vendor listing and vendor activity (activity of the whole year)

★ fully linked to GENERAL LEDGER; each invoice can be distributed to as many as five (5) different GL accounts; system automatically posts to cash and A/P accounts

ACCOUNTS RECEIVABLE

The objective of a computerized A/R system is to prepare accurate and timeley monthly statements to credit customers. Management can generate information required to control the amount of credit extended and the collection of money owed in order to maximize profitable credit sales while minimizing losses from bad debts. The programs composing this system were developed 5 years ago, especially for small businesses using the Wang Microcomputer. They have been tested in many environments since then. Each module can be used stand alone or can feed General Ledger for a fully integrated system.

CAPABILITIES:

★ menu driven; easy to use; full screen prompting and cursor control
★ invoice oriented; invoices can be entered before ready for billing, when ready for billing, after billing or after paid
★ allows entry of new invoice, credit memo, debit memo, or change/delete invoice
★ allows for progress payment
★ transaction information includes:
- type of A/R transaction
- customer P.O. #
- description of P.O.
- billing date
- general ledger account number
- invoice amount
- shipping/transportation charges
- tax charges
- payment
- progress payment information
- transaction print & file maintenance procedures insure accuracy

★ customer statements printed; computer statements with your compay letterhead can be purchased from SBSG
★ reports include: (samples on back)
- listing of invoices not yet billed
- open items (unpaid invoices)
- closed items (paid invoices)
- aging

★ fully linked to General Ledger; will post to applicable accounts; debit A/R, credits account you specify

✓ Reader Service—see page 242

H&E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

MICROSOFT BASIC COMPILER

With TRS-80™ BASIC Compiler, your Level II programs will run at record speeds! Compiled programs execute an average of 3-10 times faster than programs run under Level II. Make extensive use of integer operations, and get speeds 20-30 times faster than the interpreter.

Best of all, BASIC Compiler does it with BASIC, the language you already know. By compiling the same source code that your current BASIC interprets, BASIC Compiler adds speed with a minimum of effort.

And you get more BASIC features to program with, since features of Microsoft's Version 5.0 BASIC interpreter are included in the package. Features like the WHILE...WEND statement, long variable names, variable length records, and the CALL statement make programming easier. An exclusive BASIC Compiler feature lets you call FORTRAN and machine language subroutines much more easily than in Level II.

Simply type in and debug your program as usual, using the BASIC interpreter. Then enter a command line telling the computer what to compile and what options to use.

Voila! Highly optimized, Z-80 machine code that your computer executes in a flash! Run it now or save it for later. Your compiled program can be saved on disk for direct execution every time.

Want to market your programs? Compiled versions are ideal for distribution. You distribute only the object code, not the source, so your genius stays fully protected.

BASIC Compiler runs on your TRS-80™ Model I with 48K and disk drive. The package includes BASIC Compiler, linking loader and BASIC library with complete documentation **$195.00**

1980 INCOME TAX PAC

Completely Revised · Latest Tax Tables · Fully Tested · Complete Manual and Documentation. The new version of the Income Tax Pacs are full of error catching codes making it impossible to make an error. Follow the simple Step By Step procedure that makes tax preparation simple.

INCOME TAX PAC A .... **($19.95...Cassette)**
For Level II 16K Cassette Only
Does Form 1040 and 1040A
- Schedule A itemized deductions
- Schedule B interest and dividends
- Output to video display
- Schedule TC tax computation

INCOME TAX PAC B .... **$49.95...Cassette or Diskette)**
For Level II 16K with or without printer...cassette or disk has all features of Income Tax Pac A **Plus** works with or without line printer.
- Formats Form 1040 and 1040A for standard tax forms
- Schedule C income from a personally owned business
- Form 2106 employee business expense

PROFESSIONAL INCOME TAX PAC C .... **$99.95...Diskette**
For Level II 32K with disk and printer (optional)
Has all features of Income Tax Pac B **Plus** automatic memory storage for income tax preparers.
- 22 additional schedules and forms
- Formats forms for individual or tractor feed printing

MOD II CPA VERSION **$199.95**

GUARANTEED PROFIT 91% WINS PLACES SHOWS 32% AVERAGE PROFIT AT ALL TRACKS-1978

THE HORSE SELECTOR II (FLATS) (By Dr. Hal Davis **$50.00**
New simplified version of the original Horse Selector. The first Horse Selection System to actually calculate the estimated odds of each horse.
HIGHER PROFITS (OVER 100%) POSSIBLE THROUGH SELECTIVE BETTING ON:
- Rates each horse in 10 seconds.
- Easy to follow rules.
- Can be used with any Apple II Computer.
- 100% money back guarantee (returned for any reason).
- Uses 4 factors (speed rating, track variant, distance of the present race, distance of the last race).
- Using the above factors, the Horse Selector calculates the estimated odds. BET on horses whose actual payoff (from the Tote Board or Morning Lines) is higher than payoff based on estimated odds.
- Using the above factors, the Horse Selector calculates the estimated odds. BET on any selected horse with an estimated payoff (based on Tote Board or Morning Lines) higher than calculated payoff (based on Horse Selector II).
- Source listing for the TRS-80™, TI-59, HP-67, HP-41, Apple and BASIC Computers.
- No computer or calculator necessary (although a calculator would be helpful for the simple division used to calculate estimated odds).

FREE Dutching Tables allows betting on 2 or more horses with a guaranteed profit.

NEWDOS/80

A **New** enhanced NEWDOS for TRS-80™ Model I for the 1980's

Apparat Inc., announces the most powerful Disk Operating System for the TRS-80™. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

NEWDOS/80 is not meant to replace the present version of NEWDOS 2. 1 which satisfies most users, but is a carefully planned upward enhancement, which significantly extends NEWDOS 2. 1's capabilities. This new member to the Apparat NEWDOS' family is upward compatible with present NEWDOS 2. 1 and is supplied on Diskette, complete with enhanced NEWDOS + utility programs and documentation.
Some of the NEWDOS/80 features are:
- New BASIC commands that supports with variable record lengths up to 4095 Bytes long.
- New BASIC commands that supports with variable record lengths up to 4095 Bytes long.
- Mix or match disk drives. Supports any track count from 18 to 80. Use 35, 40 or 77 track 5" mini disk drives or 8" disk drives, or any combination.
- A security boot-up for BASIC or machine code application programs. User never sees "DOSREADY" or "READY" and is unable to "BREAK", clear screen, or issue any direct BASIC statement including "LIST."
- New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- Enhanced and improved RENUMBER that allows relocation of subroutines.
- Powerful program chaining.
- Device hanging for routing to display and printer simultaneously.
- CDE function; simultaneous striking of the C, D and E keys will allow user to enter a mini-DOS to perform some DOS commands without disturbing the resident program.
- Upward compatible with NEWDOS 2. 1 and TRSDOS 2.3.
- Includes Superzap 3.0 and all Apparat 2.1 utilities.

.... **$149.00**

STOCK MARKET MONITOR

Galactic Software Ltd.

CASSETTE VERSION **$89.00**
DISK VERSION **$99.00**

1. The system is designed for the active "trader" not the "long term" investor, as the system is "technically" oriented.
2. For the TRS-80™ Model I, Level II, 16K or more. Available in both disk and tape versions.
3. Tracks user selected issues, in a technical system that reflects the issue's performance against the overall market.
4. Set up data is input by the user from the Standard and Poors stock guide or Value Line.
5. Daily issue data, "high", "low", "close" and "volume" are input from any newspaper containing this information.
6. Daily overall market, "volume" and "closing Dow" are also provided from a newspaper.
7. Volume and price changes of an issue, as they compare to volume an price changes of the overall market, are the basis of this system's analysis of the given issue.
8. Comparisons of the issue against itself are also done. This may allow the user to spot "unusual" activity on this issue.
9. Clear indications are given as to whether the issue is "out performing", "under performing" or "performing" with the market.
10. Complete video and printed output is provided.
11. This program is intended to be a guide to indications, and is not to be used as a sole recommendation to buy, sell or hold an issue. These decisions are the responsibility of the user and his brokerage.

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

✓9

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

24 HOUR ORDER LINE
(914) 425-1535

master charge · VISA

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software (less $3 penalty for handling)

ADD $2.00 FOR SHIPPING IN UPS AREAS
ADD $3.00 FOR C.O.D. OR NON-UPS AREAS
ADD $4.00 OUTSIDE U.S.A., CANADA & MEXICO

Can Computing Be Art?

James J. Conroy KA3FAL
57 E. Garrison St.
Bethlehem, PA 18018

For years computer enthusiasts have been experimenting with different ways of using their video screens and printers to express their artistic urges. Some of these methods have been quite sophisticated and complex. I have uncovered an amazingly simple system that allows the user to quickly produce gratifying and accurate reproductions of favorite pictures with a TRS-80.

To be fair, the radio amateurs pioneered this field with their teletypes and began the initial research into digitizing images. Known as Radio Teletype (RTTY, pronounced ritty) Art, hams have been using teletypes and other printers to transmit visual information since the 1920's.

These Rembrandts of RTTY discovered that different densities of type characters approximate the shades of light and dark found in photographic images. They often used a single character element and overtyped certain portions to achieve darker tones.

Now we have a wide variety of electronic digitizing devices which, coupled with video cameras, produce faithful copies of any picture. But if you are short on cash, here is a method that can get you started for less than ten bucks. I call it CTTY ART.

How It Is Done

The major task in reproducing a photograph digitally is measuring the density, or degree of lightness and darkness of a small portion of the picture, and assigning a printer element to designate this. That this must be done for the entire photo, section by section, and by old-fashioned methods, is enough to drive anyone bananas!

Radio Shack provides each of us with a CRT map of the graphics grid of our computers, and it is your key to computer art. Cut this page out of your Level I of II manual (be brave!), and take it down to your local photostatic copying center and ask them to produce a transparent copy of it. (You want the kind of transparency which can be used on an overhead projector.) The copy should only cost a few dollars, and, with this item in hand, you have the basis for a rapid and non-destructive method of producing CTTY ART.

Now find a suitable picture. A high contrast photo will serve best to begin with, as it presents only distinct images of light and dark. I have chosen a picture of a well known holy man from the Himalayas.

Next, place the picture on a clipboard (or other secure surface), and then set the transparent video worksheet over the image. Center your subject, and then tape the worksheet. Turn on your computer and don your artist's beret . . . you're ready to go!

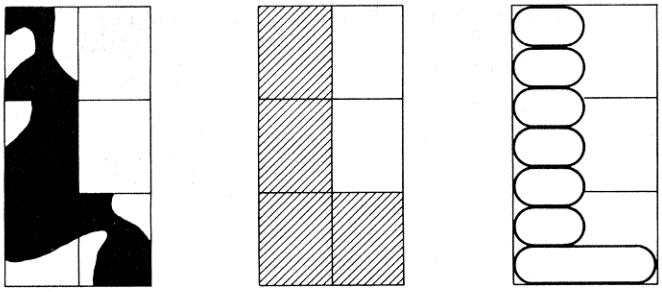
You will notice your photograph is now neatly divided into a grid of graphics blocks, and that six blocks constitute one character element. By examining the photograph (from left to right—beginning at the top) you will see that each character element block is black, white or some mixture of black and white.

All you have to do is observe this miniature pattern and decide which letter it represents most (Fig. 1).

To reproduce your art only on the screen (or your printer if it has graphics capabilities), you match the density pattern with the corresponding graphics character code. Remember—unless you have a video reverse board installed, the screen figure will appear as a negative of your original artwork.

This process is really very easy and with a few hints you will be surprised how quickly you will sail through the job. With the sequence of letters derived from your graph, create LPRINT statements. Make a note of your approximations for each line on the worksheet. If you check Program Listing 1, Add some blank lines above and below your creation and give it a go. Voila! It's fascinating to watch a recognizable image materialize before your eyes.

It does not take much work to



SAMPLE OVERLAY

CORRESPONDING CHARACTER CODE 181

CORRESPONDING LETTER L

Fig. 1

you will see how basic the program is. It is just some FOR-NEXT loops to space the top and bottom of the artwork and a loop to afford multi-copies.

Some Practical Guidelines

Here are some practical guidelines:

1. Always approximate! Sometimes the pattern may not look like any letter at all. Try punctuation marks and numbers. If that does not work, then just go for some approximate density substitute. (You can always change it later.)
2. The letter M seems to be the most dense character and, of course, a space is the lightest. The period, comma and apostrophe will serve for other light densities.
3. Use your LPRINT TAB function to place your lines properly.
4. Cut a cardboard cursor or use a ruler as a guide to keep your attention on the proper line.

After you have selected characters for each line, you can encode your LPRINT commands.

```
                  ..::HHMWHMV,,
              ,:HMMMMMMMMMMMMMWG,
           ,HRP''        'HMMMMMMMML,
         1R'               'TMMMMMMMMM,.
        ,J1                JVMMMMMMMMML,
       :HC-:-.   .JR'''9HMMMMMMMMMMMH
       JVMVVVP  .KM.''')44')MMMMMMMML1
      .JMM1      X-,.        .MMMMMMMMMML.
    6MMPP),.,,JWGWN,,...JNMMMMMMMMMM'KH.
    'YKL.)MMMR'''9MMMMMMMMMMMMMMMMMMK,.'
       )X'MMPWVWMMWW)9MMMMMMMMMMMMMMMJ('
      ''' 9, '''   ')H9MMMMMMMMMMMMMMMK
         .JL,CL,--.,,7WMMMMMMMMMMMMMM
        )L7MMLJHWWRHMMM7MMMMMMEMMMMMK'
          7H7MMMMMMM'17HHHH9MMMMM'MMMK-
           'RMMMM1   VVVYY,7M/'/VM')MM-
            'RMM7'    VVV''HY.7//    '9YM-
             'V/'      VVV''VX/'     /''
              '         VVVV'V'      /'
                         VVVV/       '
                          VV/
                          '/
```

Printout

✓ Reader Service—see page 242

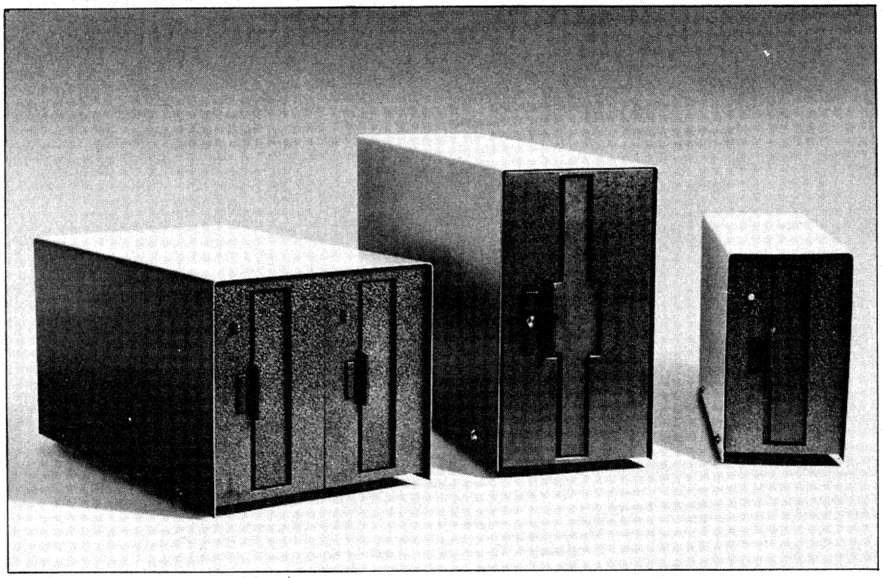
```
40 INPUT"HOW MANY COPIES?";Z:FOR C=1 TO Z
50 CLS
60 FOR Q=1 TO 10:LPRINT:NEXT Q
70 FOR A=1 TO 4:LPRINT TAB(30)".":NEXT A
80 LPRINT TAB(10)".                              .                               ."
90 LPRINT TAB(11)".                                                            ."
100 LPRINT TAB(12)".                                                          .":LPRINT
110 LPRINT TAB(24)"..::HHMWHMV,,"
120 LPRINT TAB(20)",:HMMMMMMMMMMMMMWG,"
130 LPRINT TAB(18)",HRP''     'HMMMMMMML,"
140 LPRINT TAB(17)"1R'              'TMMMMMMMM,."
150 LPRINT TAB(16)",J1              JVMMMMMMMMML,"
160 LPRINT TAB(15)":HC-:-.  .JR'''9HMMMMMMMMMMH"
170 LPRINT TAB(15)"JVMVVVP .KM.''')44')MMMMMMML1"
180 LPRINT "......        .JMM1    X-,.      .MMMMMMMMML. ¬¬¬¬
   ......"
190 LPRINT TAB(12)"6MMPP),.,,JWGWN,,...JNMMMMMMMMM'KH."
200 LPRINT TAB(12)"'YKL.)MMMR'''9MMMMMMMMMMMMMMMMMK,.'"
210 LPRINT TAB(15)")X'MMPWVWMMWW)9MMMMMMMMMMMMMMJ('"
220 LPRINT TAB(14)"''' 9, '''  ' )H9MMMMMMMMMMMMMMK"
230 LPRINT TAB(17)".JL,CL,--.,,7WMMMMMMMMMMMMM"
240 LPRINT TAB(16)")L7MMLJHWWRHMMM7MMMMMMEMMMMMK'"
250 LPRINT TAB(17)"7H7MMMMMMM'17HHHH9MMMMM'MMMK-"
260 LPRINT TAB(18)"'RMMMM1  VVVYY,7M/'/VM')MM-"
270 LPRINT TAB(19)"'RMM7'   VVV''HY.7//   '9YM-"
280 LPRINT TAB(20)"'V/'     VVV''VX/'     /''"
290 LPRINT TAB(21)"'        VVVV'v'      /'"
300 LPRINT TAB(31)"VVVV/         '"
310 LPRINT TAB(32)"VV/":LPRINT TAB(32)"'/"
320 FOR X=1 TO 10:LPRINT:NEXT X
330 LPRINT TAB(9)"H.H. MAHARISHI MAHESH YOGI  -VERSION 9.7 BY: KA3FAL
340 NEXT C
350 END
```

Program Listing 1

complete the initial version of your artwork. I was able to generate the picture of the yogi on my first effort, in much less than an hour. Since then, I have modified it many times. You will want to do this too, because the main drawback in this system is that Radio Shack's video worksheet is not printed to the exact dimensions of the video monitor. Furthermore, there will probably be even greater variance between the worksheet and your printer line and type dimensions.

Since the worksheet is narrower than the printer line, your output will be wider than the original. This can be corrected by subtracting a few characters from each line, and will vary from one work to another, as you see fit. I took out about six spaces (characters) to get the correct dimensions.

Once you get the hang of this simple method, you can examine and encode a picture into BASIC lines as you go along. Give it a try. ■



More power to you.

Disk drives, hardware and software— now more affordable *and* more available!

NEW PRODUCT ANNOUNCEMENT:
NEW MODEL III WITH DISK DRIVES!

TRS-80™ Model III with dual MPI B91 80-track disk drives—704K of reliable disk storage on only two drives!
This system features:
- Two MPI B91 double-density 80 track drives
- 48K RAM, double-density Newdos 80
- 3 millisecond access time
- Includes A.M. Electronics' controller board and MAKE80© program
- Complete, ready-to-run

Special Introductory Price **$2295**

The Ultimate Small Business Computer: TRS-80™ Model III with four dual-headed MPI B92 80-track disk drives!
This system features:
- Four proven MPI drives
- 48K RAM, double-density Newdos 80
- 3 millisecond access time
- 2.8 Megabytes reliable disk storage
- Includes A.M. Electronics' MAKE80© program and controller board
- Complete, ready-to-run

Special Introductory Price **$3795**

CASES AND POWER SUPPLIES

5½-inch enclosures
Single drive unit case and power supply **$85**
Dual drive unit case and power supply **$120**
(Extender cables are $15 each extra)

8-inch enclosures
Single drive unit case and power supply **$150**

Store Hours: Tues. - Fri. 11-7, Sat. 11-5.
Complete Service Center: For speedy repairs, including modified TRS-80's.

Attention dealers, OEM's & distributors
Call us for details on our attractive pricing.

SPECIALS!

SAVE ON COMPLETE DISK DRIVES!

5¼-inch drives
40-track MPI 51 w/case, power supply and extender cable ... **$320**
80-track MPI 91 w/case, power supply and extender cable ... **$425**
Dual-headed 80-track drive with case and power supply **$550**

8-inch drives
Single Siemens 8-inch drive with dual case and power supply .. **$695**

INTRODUCING TEAC DISK DRIVES—THE BEST DRIVES WE HAVE SEEN TO DATE!

Teac 40-track disk drive **$299**
Teac 80-track disk drive **$399**
All drives complete with cases, power supplies and extender cables.
(Add $20 for "Flippy" version)

PRINTER SPECIAL!
Epson MX-80 Printer **$495**

TRS-80™ SOFTWARE PACKAGES

MAKE80©
Converts standard 35 or 40 track diskettes into 80-track-readable diskettes ... **$14.95**

The power behind the drives©

452

A.M. ELECTRONICS, INC.
3366 Washtenaw Ave.
Ann Arbor, Michigan 48104
(313) 973-2312

Visit our retail showroom for a "hands-on" look at our TRS-80™ hardware and software

SUPER UTILITY, by Kim Watt.
Stand-alone 24K machine language program for disk includes:

ZAP UTILITY
- Read/modify data regardless of disk protection • One-step track-to-track/sector to sector • Dual cursors; ASCII and Hex. Modify in Hex, Decimal or ASCII • Display disk sectors, display file sectors, copy disk sectors, compare disk sectors, display/modify main memory, search memory or disk for specified string and return location

PURGE UTILITY
- Kill files by file spec or category • Zero out unused directories or sectors
- Compute passwords, change disk's name, date, passwords, protection levels • Directory routine indicates all active and inactive files, their location in directory and status of granules

DISK COPY UTILITY
- Copy *any* TRS-80™ readable disk, regardless of protection

TAPE COPY UTILITY
- Copy *any* TRS-80™ readable tape, regardless of protection or baud rate

DISK REPAIR UTILITY
- *Automatically* repair damaged HIT, GAT or BOOT sectors • Directory check advises of errors • Automatic recovery of killed files • Shows active and inactive files, and their location on the disk

MEMORY UTILITY
- Move memory • Jump to memory • Test memory • Compare memory • Input or output any byte to any port • Zero memory • Exchange memory • Edit memory • Load memory

SUPER UTILITY is now available for **$49.95,** plus $2.50 shipping and handling!

™TRS-80 is a trademark of Tandy Corp.

VISA

master charge

Reader Service—see page 242

You can graph any function in minutes with this spiffy routine.

Function Plotter

Louis C. Graue
624 Campbell Hill Road
Bowling Green, OH 43402

Here is a graphing program written for a TRS-80 Level II CRT display which relieves you of a good many frustrations. Its features include:

1. An input routine for a new function.
2. One value of X.
3. A graph that appears right side up, neatly framed, with the point you picked in the exact center, labeled with its coordinates.
4. No scaling distortions.
5. An ability to ignore points at which the function is undefined and to plot the rest without aborting the program.
6. Automatic choice of a sequence of central points and movement of the four by four frame along the graph.

Instead of trying to use scaling factors to distort the graph to fit the available space, we leave the graph alone, center a four by four frame on the point (A,F(A)) and display that portion of the graph. By changing the value A, we can examine any part of the graph.

Though the TRS-80 does not compute odd roots of negative numbers, there is a way around this difficulty. For example, instead of entering the function Y = (X − 1)↑(1/5) use Y = SGN(X − 1)*(SGN(X − 1)*(X − 1)↑(1/5). This does compute and gives the correct values.

If you want the program to automatically choose a sequence of central points make the following modifications:

```
70 INPUT"CENTRAL VALUE OF X";K
75 FOR A=K-5 TO K+5 STEP .1
250 FOR N=1 TO 1000:NEXT:NEXT
```

The starting and stopping points, the step size and the length of time a frame remains on display can be selected by modifying the numbers given.

With this program you will be able to enter any function and find its graph in a matter of minutes. ■

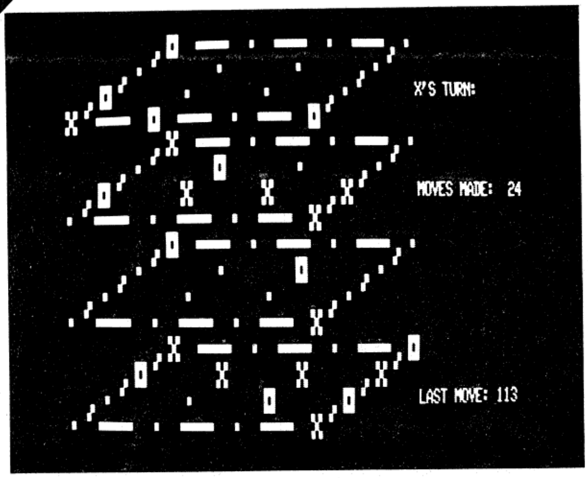
```
20 REM: PLOTS A 4 BY 4 NEIGHBORHOOD OF ANY SELECTED POI
    NT.
30 CLS
40 PRINT"PRESS THE 'H' KEY THEN ENTER 'Y=F(X)', WHERE F
    (X) IS
   YOUR FUNCTION EXPRESSED IN TERMS OF X.
     THEN ENTER 'RUN 60'
   -IF YOU WANT TO RUN THE S
    SAME FUNCTION AGAIN ENTER 'RUN 60'-
   -TO ENTER
    W FUNCTION ENTER 'RUN'-
50 EDIT 260
60 PRINT
70 INPUT"CENTRAL VALUE OF X";A
80 CLS
90 FOR I=0 TO 127    'CREATE FRAME
100 SET(I,0):SET(I,47):NEXT
110 FOR I=0 TO 47
120 SET(0,I):SET(127,I):NEXT
130 FOR J=1 TO 46 STEP 45:FOR I=1 TO 7:SET(16*I,J):NEXT
    :NEXT
140 FOR I=1 TO 126 STEP 125:FOR J=1 TO 7:SET(I,6*J-1):N
    EXT:NEXT
150 ON ERROR GOTO 280
160 X=A:GOSUB 260
170 Y1=Y
180 FOR X=A-2 TO A+2 STEP 4/126   'START PLOTTING FCN HE
    RE
190 GOSUB 260
200 L=47-(Y-Y1+2)*(47/4)
210 IF L<0 OR L>47 GOTO 230
220 SET(((126/4)*(X-A+2)),L)
230 NEXT X
240 PRINT@543,"(";A;",";Y1;")";
250 GOTO 250
260 Y=SIN(3*X+1)*COS(5*X)
270 RETURN
280 IF ERR/2+1=5 OR ERR/2+1=11 RESUME NEXT


DONE
```

Program Listing

FOR TRS-80*

QUAD



by Charles Asper

Tic-Tac-Toe, *FOUR* in a row? That's only one of the differences with Quad, Acorn's challenging three dimensional game. A graphically represented cube, four rows by four columns by four levels, Quad gives you 64 different playing positions for your X's and O's, and 76 different winning four-in-a-row combinations.

Play Quad against your computer or a friend; against the computer, there are four levels of difficulty available. You can rotate the cube six different ways to get a new perspective (or confuse your opponent!). For even more challenge, set the built-in game clock — it'll forfeit your move if time runs out!

Available for Level II, 16K. $14.95 for tape, $20.95 on disk.

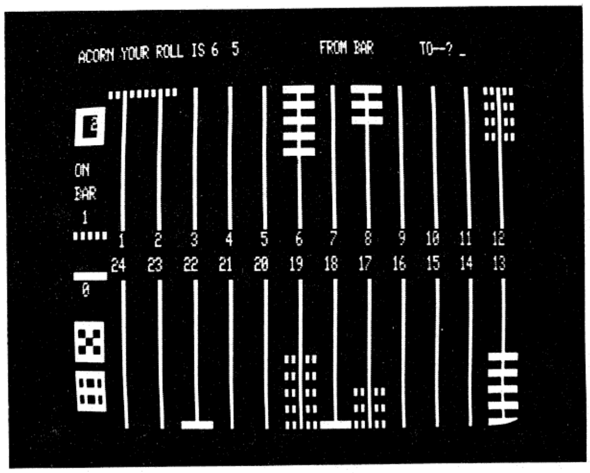
These are just two of Acorn's wide selection of game, utility, educational and business programs for the TRS-80*.

Acorn ✓34
Software Products, Inc.

634 North Carolina Avenue, S.E., Washington, D.C. 20003

NEW, SEMI-COMPILED VERSION!

GAMMON CHALLENGER



by Ray Daly & Tom Throop

The backgammon player featured in *Personal Computing* (August 1979) is now back in a faster, even better version! The game logic of the new Gammon Challenger has been compiled to machine language for extra speed, and there are more special features than ever.

Choose one of three levels of play, but don't get too ambitious — Gammon Challenger will put your skill to the test at all levels. For serious players, the "doubling cube" option can be used for added excitement. There *are* other computer backgammon games, but none quite like Gammon Challenger. Supplied on tape for $14.95. Requires Level II, 16K.

*TRS-80 is a trademark of Tandy Corp.

These and other popular Acorn programs are available now at fine computer stores. Ask for them.

DEALER INQUIRIES INVITED

Get your message across with a micro generated banner.

Banner Banter

Valerie Vann
631 G St.
Davis, CA 95616

Radio Shack's Quick Printer II will squeeze into a tight budget (under $220), while affording a kind of hard copy fun that you probably thought required a full-feature printer.

The Silver Ribbon Banner Printer is an interactive Level II BASIC 16K program which prints banners on the Quick Printer II and other printers.

Write a "Happy Anniversary" banner to present to your spouse. Show your friends that the "funny looking paper" your new printer uses is really high class stuff! Or print certificates of merit for your students when they complete a session of CAI (Computer Assisted Instruction).

When I began this program, I planned to use a simple 7 × 9 dot matrix character set. But 15 years as a graphic artist overcame my good intentions. My custom-designed typeface stretched the program out, instead, to 300 lines. I learned a lot about Level II BASIC string-handling functions in the process.

Five Messages

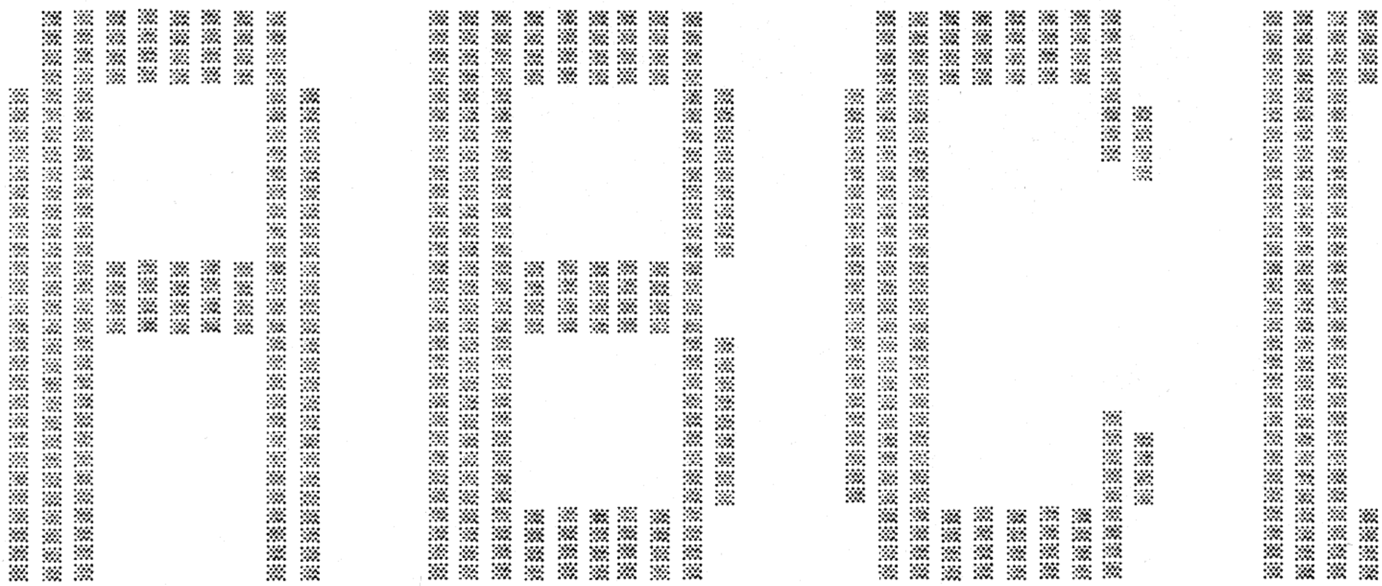
The program gives you the option of printing five messages, like "Happy Birthday," or your own creation of 30 characters or less.

The banner characters are printed with a solid black graphics block, one of the special characters which can be printed by specifying the character code, in this case CHR$(127).

For printing, the input message string (C$) is taken apart one character at a time, working from left to right. First, the ASCII code of the leftmost character is determined by the ASC(C$) instruction in line 56. Then, a set of IF C = n THEN m and ON n GOTO m instructions routes the program to the appropriate subroutine for that character (N = number computed from ASCII code, m = subroutine line number). See lines 57–60 and 66–72.

Each banner character is formed by printing a number of lines (usually 10) from a combination of 30 blocks and/or spaces. In the subroutine, each line is put together in a single string by concatenating elements (adding strings together) of a set of string constants of either blocks or spaces.

These constants are defined in lines 5-14. Subroutines for a few characters such as Q, I, and

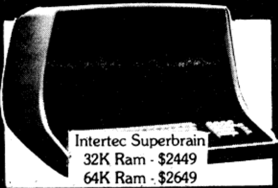


Sample 1. Base 2 Banner

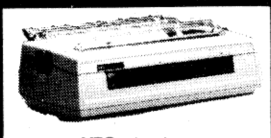
ΩMEGA SALES CO. ✓389

"WHOLESALE COMPUTER PRICES" DIRECT TO THE PUBLIC

12 Meeting St., Cumberland, RI 02864

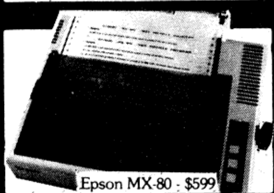


Intertec Superbrain
32K Ram - $2449
64K Ram - $2649



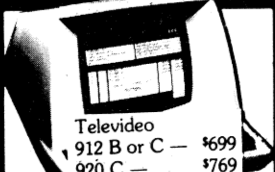
NEC spinwriter
5510-5530 - $2449

Apple II - 16K $ 949
48K $1099



Epson MX-80 - $599

Atari 800 - $769



Televideo
912 B or C — $699
920 C — $769

PRODUCT SPECIAL of the MONTH!!

Diablo 630
$1995
$2195 (with tractor feed)

Products are NOW IN STOCK AT ΩMEGA Sales Co.

Centronics
(limited quantities-call for availability)

779-II	$ 749
704-9	$1500
730-3	$ 599
737-1	$ 699

ΩMEGA OFFERS THE BEST DELIVERY AND PRICE ON:
APPLE • ATARI • TRS-80 Model II • INTERTEC • DIABLO • EPSON • HEWLETT—PACKARD • SOROC • COMMODORE • NEC • QUME • CENTRONICS

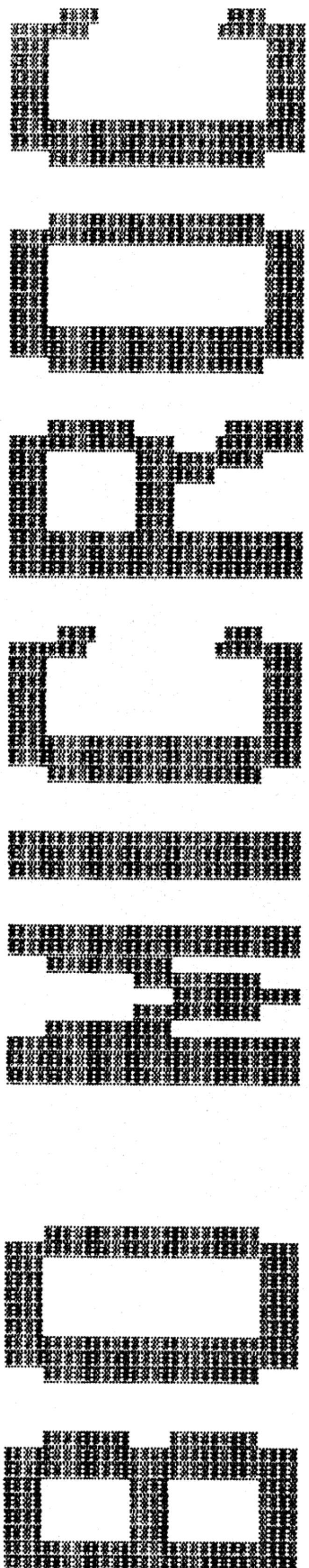
CALL TOLL FREE FOR ΩMEGA'S PRICE!

1-800-556-7586

ΩMEGA sells only quality merchandise to our customers.
ΩMEGA will try to match any current advertised price with similar purchase conditions.
Before you buy anywhere else — be sure to call ΩMEGA Sales Co.
1-401-722-1027

ΩMEGA "A member in good standing of the better business bureau."

ΩMEGA ships via UPS, truck, or air. COD's.
Visa, Mastercharge accepted, with no service charge.



Sample 2. Base 2 Banner (condensed version)

? provide variations on the 10-line format or the basic string constant building block set.

As each line string is built up, it is put into a string array, L$(10). Then the printing subroutine, line 296, outputs the array to the printer, line by line, to print the banner character.

The character just printed is then dropped from the original message string using the RIGHT$(C$,n) instruction, line 75, and the process is repeated for the new left character.

The program could be adapted to run on small printers which use narrow roll paper, like the Quick Printer I. You may wish to make the characters taller or shorter by changing the length of the lines. Or, choose a different character if the 'block' isn't available.

Words of Caution

If your message contains illegal characters in the middle, the program will print out all the legal characters up to that point. Then, it will abort and send you back to the instructions to input your message again. This wastes paper.

You can also go through lots of paper if you turn the kids loose. The Quick Printer II is supposed to have a print-head life of 30 million impressions, but the half-life point could arrive in a hurry if you don't exercise restraint with paper.

Base 2 Printer Banners

Sample 1 is printed on the Base 2 Model 800 MST Printer with no modification to the program. It uses the regular TRS-80 configuration of the printer, and 80 characters per line at six lines per inch. The character CHR$(127) is a graphics-type block on the Base 2. It is made up of every other dot of the print head matrix.

The printer is RESET before the program runs to make sure that the AUTO form feed is disabled. The banner characters are tall rather than wide, as on the QP-II.

Sample 2 also has tall characters, but denser printing. The configuration of the Base 2 printer was altered by typing in the following line before running the program:

```
LPRINT CHR$(27); CHR$(74); CHR$(27);
       CHR$(98); CHR$(14); CHR$(27);
       CHR$(106); CHR$(27); CHR$(52);
```

This sequence of command characters to the printer does a RESET, and sets the vertical spacing to 14 half-dots (no space between lines). It also enables uni-directional printing to improve vertical alignment of the characters, and sets the horizontal spacing to 132 characters per line. If the Base 2 is to be used regularly to print banners, this line should be added to the beginning of the program.

The Base 2 printer has no side margin controls, so if you don't want your banner printed along the left margin of the paper, you can put a TAB(10) in front of each LPRINT statement in the program. Or you can add a string of blanks (B3$, line 8, for example) to the front of each printed line of blocks. These changes need to be made to the main printing subroutine, lines 296-298, and also to the subroutines for printing the characters 1, I, Q and !. These subroutines begin with a REM statement and are located at lines 87, 182, 221, and 281, respectively. ■

Program Listing. Banner Printer

```
1 CLS:PRINT"TRS-80  QUICK PRINTER II   BANNER PRINTER"
2 PRINT"BY  VALERIE VANN, DAVIS,CALIF, 4 OCT 79"
3 FOR X=1TO1800:NEXTX
4 CLEAR 1000:DIM L$(10)
5 B0$=STRING$(1,32)
6 B1$=STRING$(4,32)
7 B2$=STRING$(9,32)
8 B3$=STRING$(13,32)
9 B4$=STRING$(26,32)
10 L1$=STRING$(4,127)
11 L2$=STRING$(9,127)
12 L3$=STRING$(13,127)
13 L4$=STRING$(26,127)
14 L5$=STRING$(30,127)
```

Program continues

✓Reader Service—see page 242

```
15 CLS:PRINT@32Ø,"PLEASE MAKE SURE THAT YOUR QUICK PRIN
   TER II"
16 PRINT"HAS PLENTY OF PAPER IN IT,"
17 PRINT"AND THAT IT IS  * ON LINE *  AND READY TO PRIN
   T."
18 FOR X=1TO27ØØ:NEXTX
19 CLS:PRINT"CHOOSE THE MESSAGE YOU WISH TO PRINT FROM
   THE LIST."
2Ø PRINT"TYPE THE NUMBER OF THE MESSAGE AND PRESS  ENTE
   R.":PRINT
21 PRINT"HAPPY BIRTHDAY        (1)"
22 PRINT"HAPPY ANNIVERSARY     (2)"
23 PRINT"HAPPY NEW YEAR        (3)"
24 PRINT"MERRY CHRISTMAS       (4)"
25 PRINT"HAPPY CHANUKAH        (5)"
26 PRINT"WRITE YOUR OWN        (6)"
27 INPUT A
28 IF1<=A AND A<=6 THEN 31
29 CLS:PRINT@45Ø,"INCORRECT INPUT. PLEASE LOOK AT THE L
   IST AGAIN."
3Ø FORX=1TO1ØØØ:NEXTX:GOTO19
31 ON A GOTO 33,34,35,36,37,38
32 GOTO29
33 C$="HAPPY BIRTHDAY":GOTO48
34 C$="HAPPY ANNIVERSARY":GOTO48
35 C$="HAPPY NEW YEAR":GOTO48
36 C$="MERRY CHRISTMAS":GOTO48
37 C$="HAPPY CHANUKAH":GOTO48
38 CLS:PRINT"YOUR MESSAGE MAY BE 3Ø CHARACTERS LONG, CO
   UNTING SPACES"
39 PRINT"ONLY UPPER CASE LETTERS - - A THRU Z - -"
4Ø PRINT"              AND NUMBERS - - Ø THRU 9 - -"
41 PRINT"     AND THE CHARACTERS - - !  OR  ? - -ARE AL
   LOWED."
42 PRINT:PRINT"DO NOT USE COMMAS OR OTHER SYMBOLS.":PRI
   NT
43 PRINT:PRINT"* * DO NOT PUT BLANKS IN FRONT OF YOUR M
   ESSAGE * *":PRINT
44 Q$=CHR$(92)
45 PRINT"TYPE YOUR MESSAGE NOW.  THEN PRESS  ENTER.":PR
   INT"YOUR MESSAGE SHOULDN'T GO PAST THE END OF THE
   LINE BELOW ";Q$
46 PRINT STRING$(32,"-")
47 INPUTC$
48 LN=LEN(C$):LE=LN
49 IF LN<=3Ø THEN 53
5Ø CLS:PRINT"TOO MANY CHARACTERS."
51 PRINT"PLEASE TRY AGAIN."
52 FORX=1TO1ØØØ:NEXTX:GOTO38
53 E=Ø:LPRINT:LPRINT:LPRINT:LPRINT:LPRINT
54 FORZ=1TO LE
55  IF E<>Ø THEN 76
56  C=ASC(C$)
57  IF C=32 THEN 63
58 IFC=33THEN281
59  IF 48<=C AND C<=57 THEN 64
6Ø IFC=63THEN288
61  IF 65<=C AND C<=96 THEN 69
62  E=1:GOTO76
63 FOR T=1TO6:LPRINT:NEXTT:GOTO74
64  C=C-47
65  IF C=2 THEN 87
66  ON C GOSUB 82,87,91,98,1Ø3,11Ø,116,123,13Ø,135
67  GOSUB296
68  GOTO74
69  C=C-64
7Ø IF C=9 THEN 182
71 IF C=17 THEN 221
72  ON C GOSUB 14Ø,145,15Ø,155,16Ø,165,17Ø,177,182,186,
    191,198,2Ø3,2Ø8,214,216,221,228,234,24Ø,245,25Ø,25
    6,262,268,274
73  GOSUB296
74 LN=LN-1
75  C$=RIGHT$(C$,LN)
76 NEXT Z
77 IF E<>Ø THEN 81
78 FORR=1TO6
79  LPRINT:NEXTR
8Ø END
81 CLS:PRINT"ILLEGAL CHARACTERS":GOTO51
82 REM Ø
83 L$(1)=B1$+L3$+L2$:L$(1Ø)=L$(1)
84 L$(2)=L5$:L$(3)=L5$:L$(9)=L5$
85 FORX=4TO8:L$(X)=L1$+B3$+B2$+L1$:NEXTX
86 RETURN
87 REM 1
88 LPRINT:LPRINT B2$;B3$;L1$
89 FORX=1TO3:LPRINT L5$:NEXTX
9Ø LPRINT:LPRINT:GOTO74
91 REM 2
92 L$(1)=B1$+L2$
93 L$(2)=L1$+L3$+B2$+L1$:L$(3)=L1$+L3$+B2$+L1$
94 FORX=4TO7:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
95 L$(8)=L1$+B2$+L3$+L1$:L$(9)=L1$+B2$+L3$+L1$
96 L$(1Ø)=L1$+B2$+L3$+L1$
97 RETURN
98 REM 3
99 L$(1)=L1$+B2$+B3$+L1$:L$(2)=L$(1)
1ØØ FORX=3TO7:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
1Ø1 L$(8)=L5$:L$(9)=L5$:L$(1Ø)=B1$+L2$+B1$+L2$+B1$
```

Program continues

```
1Ø2 RETURN
1Ø3 REM 4
1Ø4 L$(1)=B3$+L1$+B3$:L$(2)=B3$+L3$
1Ø5 L$(3)=B3$+L3$+L1$
1Ø6 FORX=4TO6:L$(X)=B3$+L1$:NEXTX
1Ø7 L$(7)=L5$:L$(8)=L5$:L$(9)=L5$
1Ø8 L$(1Ø)=B3$+L1$
1Ø9 RETURN
11Ø REM 5
111 FORX=1TO3:L$(X)=L1$+B2$+L1$+L3$:NEXTX
112 FORX=4TO7:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
113 L$(8)=L1$+L3$+B2$+L1$:L$(9)=L1$+L3$+B2$+L1$
114 L$(1Ø)=B1$+L2$
115 RETURN
116 REM 6
117 L$(1)=B1$+L2$+L3$+B1$
118 L$(2)=L5$:L$(3)=L5$
119 FORX=4TO7:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
12Ø L$(8)=L1$+L3$+B2$+L1$
121 L$(9)=L1$+L3$+B2$+L1$:L$(1Ø)=B1$+L2$
122 RETURN
123 REM 7
124 L$(1)=B2$+B2$+B1$+L1$
125 FORX=2TO4:L$(X)=B4$+L1$:NEXTX
126 L$(5)=L2$+B3$+B1$+L1$:L$(6)=L3$+B3$+L1$
127 L$(7)=L5$:L$(8)=B1$+B1$+L2$+L3$
128 L$(9)=B3$+B1$+L3$:L$(1Ø)=B4$+L1$
129 RETURN
13Ø REM 8
131 L$(1)=B1$+L2$+B1$+L2$:L$(1Ø)=L$(1)
132 L$(2)=L5$:L$(3)=L5$:L$(8)=L5$:L$(9)=L5$
133 FORX=4TO7:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
134 RETURN
135 REM 9
136 L$(1)=B1$+B3$+L2$:L$(2)=L1$+B2$+L3$+L1$:L$(3)=L$(2)
137 FORX=4TO7:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
138 L$(8)=L5$:L$(9)=L5$:L$(1Ø)=B1$+L3$+L2$
139 RETURN
14Ø REM A
141 L$(1)=L4$:L$(1Ø)=L4$
142 L$(2)=L5$:L$(3)=L5$:L$(9)=L5$
143 FORX=4TO8:L$(X)=B3$+L1$+B2$+L1$:NEXTX
144 RETURN
145 REM B
146 L$(1)=L5$:L$(2)=L5$:L$(3)=L5$:L$(9)=L5$
147 FORX=4TO8:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
148 L$(1Ø)=B1$+L2$+B1$+L2$
149 RETURN
15Ø REM C
151 L$(1)=B1$+L2$+L3$:L$(2)=L5$:L$(3)=L5$
152 FORX=4TO8:L$(X)=L1$+B2$+B3$+L1$:NEXTX
153 L$(9)=L2$+B1$+B2$+L1$+L1$:L$(1Ø)=B1$+L1$+B3$+L1$
154 RETURN
155 REM D
156 FORX=1TO3:L$(X)=L5$:NEXTX
157 FORX=4TO8:L$(X)=L1$+B2$+B3$+L1$:NEXTX
158 L$(9)=L5$:L$(1Ø)=B1$+L2$+L3$
159 RETURN
16Ø REM E
161 FORX=1TO3:L$(X)=L5$:NEXTX
162 FORX=4TO8:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
163 L$(9)=L1$+B2$+B3$+L1$:L$(1Ø)=L$(9)
164 RETURN
165 REM F
166 FORX=1TO3:L$(X)=L5$:NEXTX
167 FORX=4TO8:L$(X)=B3$+L1$+B2$+L1$:NEXTX
168 L$(9)=B4$+L1$:L$(1Ø)=L$(9)
169 RETURN
17Ø REM G
171 L$(1)=B1$+L2$+L3$:L$(2)=L5$:L$(3)=L5$
172 FORX=4TO6:L$(X)=L1$+B2$+B3$+L1$:NEXTX
173 L$(7)=L1$+B2$+L1$+B2$+L1$:L$(8)=L$(7)
174 L$(9)=L1$+L3$+B1$+B1$+L1$
175 L$(1Ø)=B1$+L3$+B1$+B1$+L1$
176 RETURN
177 REM H
178 FORX=1TO3:L$(X)=L5$:NEXTX
179 FORX=4TO8:L$(X)=B3$+L1$:NEXTX
18Ø L$(9)=L5$:L$(1Ø)=L5$
181 RETURN
182 REM I
183 LPRINT
184 FORX=1TO3:LPRINTL5$:NEXTX
185 LPRINT:LPRINT:GOTO74
186 REM J
187 L$(1)=B1$+L1$:L$(2)=L2$:L$(3)=L2$
188 FORX=4TO7:L$(X)=L1$:NEXTX
189 L$(8)=L5$:L$(9)=L5$:L$(1Ø)=B1$+L4$
19Ø RETURN
191 REM K
192 FORX=1TO3:L$(X)=L5$:NEXTX
193 FORX=4TO6:L$(X)=B3$+L1$:NEXTX
194 L$(7)=B2$+L3$:L$(8)=B1$+L2$+B1$+L2$
195 L$(9)=L2$+B1$+B1$+B1$+L2$
196 L$(1Ø)=L1$+L1$+BØ$+B3$+L1$+L1$
197 RETURN
198 REM L
199 FORX=1TO3:L$(X)=L5$:NEXTX
2ØØ FORX=4TO8:L$(X)=L1$:NEXTX
2Ø1 L$(9)=L1$+L1$:L$(1Ø)=L$(9)
2Ø2 RETURN
```

Program continues

```
203 REM M
204 FORX=1TO3:L$(X)=L5$:NEXTX
205 L$(4)=B3$+L3$:L$(5)=B1$+L3$:L$(6)=L3$
206 L$(7)=B1$+L3$:L$(8)=B3$+L3$:L$(9)=L5$:L$(10)=L5$
207 RETURN
208 REM N
209 FORX=1TO3:L$(X)=L5$:NEXTX
210 L$(4)=B3$+B1$+L2$:L$(5)=B3$+L2$
211 L$(6)=B2$+L2$:L$(7)=B1$+L2$
212 L$(8)=L2$:L$(9)=L5$:L$(10)=L5$
213 RETURN
214 GOSUB 82
215 RETURN
216 REM P
217 FORX=1TO3:L$(X)=L5$:NEXTX
218 FORX=4TO8:L$(X)=B3$+L1$+B2$+L1$:NEXTX
219 L$(9)=B3$+L3$+L1$:L$(10)=B3$+B1$+L2$
220 RETURN
221 REM Q
222 L$(1)=B1$+L2$+L3$:L$(2)=L5$:L$(3)=L5$:L$(4)=L1$+B3$
    +B2$+L1$
223 L$(5)=L$(4):L$(6)=L3$+B2$+B1$+L1$
224 L$(7)=L1$+B2$+L1$+B2$+L1$:L$(8)=L$(7):L$(9)=L5$
225 FORX=1TO9:LPRINT L$(X):NEXTX
226 LPRINT L$(1):LPRINT L2$:LPRINT L1$:LPRINT L1$
227 LPRINT:LPRINT:GOTO74
228 REM R
229 FORX=1TO3:L$(X)=L5$:NEXTX
230 FORX=4TO6:L$(X)=B3$+L1$+B2$+L1$:NEXTX
231 L$(7)=B2$+L1$+L1$+B2$+L1$:L$(8)=B1$+L3$+B2$+L1$
232 L$(9)=L2$+B1$+L1$+L3$:L$(10)=L1$+L1$+B2$+L2$
233 RETURN
234 REM S
235 L$(1)=B1$+L1$+B2$+L2$:L$(2)=L2$+B1$+L1$+L3$:L$(3)=L
    1$+B2$+L1$+L3$
236 FORX=4TO7:L$(X)=L1$+B2$+L1$+B2$+L1$:NEXTX
237 L$(8)=L3$+L1$+B2$+L1$:L$(9)=L3$+L1$+B1$+L2$
238 L$(10)=B1$+L2$+B2$+L1$
239 RETURN
240 REM T
241 FORX=1TO4:L$(X)=B4$+L1$:NEXTX
242 FORX=5TO7:L$(X)=L5$:NEXTX
243 FORX=8TO10:L$(X)=L$(1):NEXTX
244 RETURN
245 REM U
246 L$(1)=B1$+L4$:L$(10)=L$(1)
247 L$(2)=L5$:L$(3)=L5$:L$(9)=L5$
248 FORX=4TO8:L$(X)=L1$:NEXTX
249 RETURN
250 REM V
251 L$(1)=B3$+B1$+L3$:L$(2)=B3$+L1$+L3$
252 L$(3)=B2$+L1$+L1$+L3$:L$(4)=B1$+B1$+L3$:L$(5)=B1$+L
    2$
253 L$(6)=L1$+L1$:L$(7)=L$(5):L$(8)=B2$+L3$
254 L$(9)=B3$+L3$:L$(10)=B1$+B3$+L3$
255 RETURN
256 REM W
257 L$(1)=B3$+L1$+L3$:L$(2)=B2$+L1$+L1$+L3$
258 L$(3)=B1$+L4$:L$(4)=L2$+L2$:L$(5)=B1$+L2$
259 L$(6)=B1$+B1$+L3$:L$(7)=L$(5)
260 L$(8)=L$(4):L$(9)=L$(2):L$(10)=L$(1)
261 RETURN
262 REM X
263 L$(1)=L2$+B1$+B1$+B1$+L2$:L$(2)=L$(1):L$(10)=L$(1)
264 L$(3)=L3$+B1$+L3$:L$(9)=L$(3)
265 L$(4)=B2$+L1$+B1$+L1$:L$(8)=L$(4)
266 L$(5)=B3$+L1$:L$(6)=L$(5):L$(7)=L$(5)
267 RETURN
268 REM Y
269 L$(1)=B3$+B1$+L3$:L$(10)=L$(1)
270 L$(2)=B3$+L1$+L3$:L$(3)=L$(2):L$(9)=L$(2)
271 L$(4)=B3$+L1$:L$(8)=L$(4)
272 L$(5)=L3$+L1$:L$(6)=L$(5):L$(7)=L$(5)
273 RETURN
274 REM Z
275 L$(1)=L1$+B3$+B2$+L1$:L$(2)=L1$+L1$+B2$+B2$+L1$
276 L$(3)=L2$+B3$+B1$+L1$:L$(4)=L3$+B3$+L1$
277 L$(5)=L1$+B1$+L2$+B2$+L1$:L$(6)=L1$+B2$+L2$+B1$+L1$
278 L$(7)=L1$+B3$+L3$:L$(8)=L1$+B1$+B3$+L2$
279 L$(9)=L1$+B2$+B2$+L1$+L1$:L$(10)=L1$+B2$+B3$+L1$
280 RETURN
281 REM !
282 LPRINT:LPRINT:L$(1)=B1$+B3$+B1$+L2$
283 L$(2)=L1$+B0$+B1$+L1$+L1$+L3$
284 L$(3)=L1$+B0$+L1$+L1$+L1$+L3$
285 LPRINT L$(1):LPRINT L$(2):LPRINT L$(3)
286 LPRINT L$(2):LPRINT L$(1):LPRINT:LPRINT
287 GOTO74
288 REM ?
289 L$(1)=B1$+B1$+B3$+L2$:L$(2)=L$(1):L$(3)=L$(1)
290 L$(5)=L1$+B0$+L1$+L1$+L1$+B2$+L1$:L$(4)=L1$+B0$+B1$
    +L1$+L1$+B2$+L1$
291 L$(6)=L$(4):L$(7)=B3$+L1$+B2$+L1$
292 L$(8)=B3$+L1$+L3$:L$(9)=L$(8)
293 L$(10)=B1$+B3$+L2$:LPRINT
294 GOSUB296
295 GOTO74
296 LPRINT
297 FORY=1TO10
298  LPRINT L$(Y):NEXTY
299 LPRINT:LPRINT:RETURN
```


Reader Service—see page 242

Train your printer to jump through loops.

Paper Tiger Screenprinter

Ruth Lewart
12 Georjean Drive
Holmdel, NJ 07733

If you want hard copy of the graphics displays you've created on your TRS-80 screen but don't want to bother with camera and tripod, you can get high-quality reproductions with a graphics printer.

The Level II Screenprinter routine, which merges with an existing program, harnesses the Integral Data Systems IDS-440 Paper Tiger (or similar graphics printer) to produce exact copies of what appears on your screen, whether graphics, alphanumeric, or a mixture of the two.

Writing the Program

While writing a program to print an alphanumeric screen or a pure graphics screen is easy, writing one to handle a mixture of alphanumerics and graphics on the screen is no trivial matter.

Complications arise because the Paper Tiger operates in two distinct modes: alphanumeric, or graphics. Note: no such distinction is made on the screen; each character corresponds to a single ASCII byte. This is also true of the printer when it is in alphanumeric mode—one byte represents one alphanumeric character.

However, in graphics mode, one byte represents one column of dots on the printer; typically, six or more bytes are needed to represent a single character. The advantage of the graphics mode is that you can print any character you can devise, including all TRS-80 graphics.

Switching back and forth between the two modes is a simple programming task, but produces alignment problems because of the differences in horizontal and vertical sizes associated with the Paper Tiger's graphic and alphanumeric modes. The solution is to use only the graphics mode, that is, to generate both alphanumeric and graphic symbols via software, rather than via the printer's character generator. The result is a large, clean print that faithfully replicates the screen.

Making Changes

Screenprinter is written to facilitate changes. For example, the routine shown in Program Listing 1 produces large print—each row on the TRS-80 screen is represented by four rows on the printout in Fig. 1. Only five changes, as shown in Program Listing 2, produce the small-size, or three-row printout of Fig. 2. If the user wants a gothic font, for example, or a Greek alphabet, only the data statements would have to be changed.

The initialization section converts the data statements into a table describing the shape of each alphanumeric character, which is then stored in computer memory. Since all the TRS-80 graphic characters can be generated quickly by a simple algorithm, a table of graphics characters is not generated during initialization. Screen printer can then run with a minimum memory.

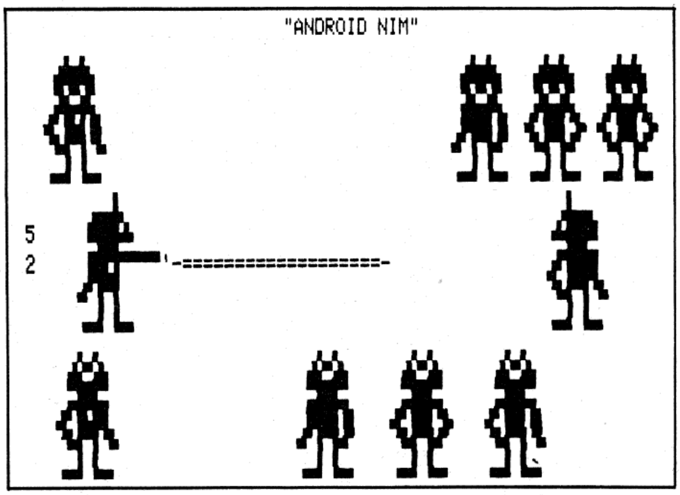


Fig. 1. Example of large-size Screenprinter merged with Android Nim

```
10000 'LARGE-SIZE SCREENPRINTER
10010 IF Q1=0 GOSUB 10500  'INITIALIZATION FOR 1ST PRIN
     T
10020 LPRINT QF$     'FORMFEED
10030 LPRINT Q2$Q3$;            'RESTORE NORMAL MODE; ENTE
     R GRAPHICS MODE
10040 LPRINT Q4$Q4$;                   'PRINT CORNERS & TOP O
     F FRAME
10050 LPRINT STRING$(200,QU$);STRING$(188,QU$);
10060 LPRINT Q4$Q4$Q3$QT$;
10070 QS=15360  'START OF VIDEO MEMORY
10080 FOR QL= 0 TO 15
10090 FOR QI=0 TO 63: QN(QI)=PEEK(QS+QI):NEXT  'READ ON
     E LINE
10100 FOR QK= 0 TO 3
10110 IF INKEY$="B" THEN LPRINT Q3$Q2$: RETURN   'LEAVE
      GRAPHICS
10120 LPRINT QL$QL$QZ$QZ$;                'PRINT LEFT SIDE O
     F FRAME
10130 FOR QI=0 TO 63
10140 IF QN(QI)>127 THEN LPRINT QG$(QM(QN(QI) AND 21),Q
     K); QG$(QM(QN(QI) AND 42),      QK);ELSE LPRINT QZ
     $;QA$(QN(QI)-32,QK);
10150 NEXT
10160 LPRINT QZ$QZ$QL$QL$Q3$QT$;    'PRINT RIGHT SIDE O
     F FRAME AND TAB
10170 NEXT QK: QS=QS+Q8: NEXT QL
10180 LPRINT Q5$Q5$;                   'PRINT BOTTOM AND CORN
     ERS
10190 LPRINT STRING$(200,QB$); STRING$(188,QB$);
10200 LPRINT Q5$Q5$Q3$QT$Q3$Q2$   'RETURN TO ALPHA MODE
10210 LPRINT QF$
10220 RETURN
10230 '
10500 'INITIALIZATION
10510 Q1=1:Q8=64:QZ$=CHR$(128)   'INITIALIZATION FLAG,
     LINE INCREMENT, GRAPHIC ZE      RO
10520 Q2$=CHR$(130):Q3$=CHR$(131):QT$=CHR$(139):QF$=CHR
     $(140) 'CONTROL CHARACTERS       FOR NORMAL, GRAPHI
     CS, TAB, AND FORMFEED
10530 Q4$=CHR$(60):Q5$=CHR$(143):QU$=CHR$(140):QB$=CHR$
     (140):QL$=CHR$(63)   'GRAP       HICS CHARACTERS FO
     R FRAME
10540 DIM QG$(7,3),QA$(64,3),QM(42),QN(64)   'GRAPHIC,
     ALPHA-NUMERIC STRINGS AND       TABLE ENTRIES
10550 FOR QQ=0 TO 7: FOR QK=0 TO 3   'CREATE GRAPHIC ST
     RINGS
10560 READ QC:QC=QC OR 128:QG$(QQ,QK)=STRING$(3,CHR$(QC
     )):IF QC=131 THEN QG$(QQ,Q      K)=STRING$(6,CHR$(
     QC))
10570 NEXT QK,QQ
10580 FOR QI=0 TO 63: FOR QJ=0 TO 4  'CREATE ALPHA STRI
     NGS
10590 READ QC:QW=(12*(QC AND 1)+24*(QC AND 2) )OR 128:
     IF QW<>131 QA$(QI,0)=QA$(Q      I,0)+CHR$(QW) ELSE
      QA$(QI,0)=QA$(QI,0)+Q3$+Q3$
10600 QW=(3*SGN(QC AND 4)+12*SGN(QC AND 8)+3*(QC AND 16
     )) OR 128: IF QW<>131 THEN      QA$(QI,1)=QA$(QI,1
     )+CHR$(QW) ELSE QA$(QI,1)=QA$(QI,1)+Q3$+Q3$
10610 QW=(3*SGN(QC AND 32)+12*SGN(QC AND 64)) OR 128:IF
      QW<>131  QA$(QI,2)=QA$(QI      ,2)+CHR$(QW) ELSE
     QA$(QI,2)=QA$(QI,2)+Q3$+Q3$
10620 NEXT QJ:QA$(QI,3)=STRING$(5,QZ$)
10630 NEXT
10640 QM(0)=0:QM(1)=1:QM(4)=2: QM(5)=3:QM(16)=4:QM(17)=
     5:QM(20)=6:QM(21)=7
10650 QM(2)=1:QM(8)=2:QM(10)=3:QM(32)=4:QM(34)=5:QM(40)
     =6:QM(42)=7    'TABLE
10660 RETURN
10670 DATA ,,,, 63,3,,, ,60,15,, 63,63,15,0
10680 DATA ,,48,63, 63,3,48,63, ,60,63,63, 63,63,63,63
10690 DATA ,,,,, ,,95,,, ,7,,7,, 20,127,20,127,20
10700 DATA 36,42,127,42,18, 35,19,8,100,98, 54,73,86,32
     ,80, ,,7,,0
10710 DATA 28,34,65,,, ,,65,34,28, 34,20,127,20,34, 8,8
     ,62,8,8
10720 DATA 0,64,48,,, 8,8,8,8,8, ,,64,,, 32,16,8,4,2
10730 DATA 62,81,73,69,62, ,66,127,64,0, 98,81,73,73,70
     , 33,65,73,77,51
10740 DATA 24,20,18,127,16, 39,69,69,69,57, 60,74,73,73
     ,49, 1,113,9,5,3
10750 DATA 54,73,73,73,54, 70,73,73,41,30, 0,,20,,0, 0,
     64,52,,0
10760 DATA 8,20,34,65,0, 20,20,20,20,20, 0,65,34,20,8,
     2,1,89,5,2
10770 DATA 62,65,93,89,78, 124,18,17,18,124, 127,73,73,
     73,54, 62,65,65,65,34
10780 DATA 127,65,65,65,62, 127,73,73,73,65, 127,9,9,9,
     1, 62,65,65,81,113
10790 DATA 127,8,8,8,127, 0,65,127,65,0, 32,64,64,64,63
     , 127,8,20,34,65
10800 DATA 127,64,64,64,64, 127,2,12,2,127, 127,4,8,16,
     127, 62,65,65,65,62
10810 DATA 127,9,9,9,6, 62,65,81,33,94, 127,9,25,41,70,
      38,73,73,73,50
10820 DATA 1,1,127,1,1, 63,64,64,64,63, 31,32,64,32,31,
      127,32,24,32,127
10830 DATA 99,20,8,20,99, 3,4,120,4,3, 97,81,73,69,67,
     4,2,127,2,4
10840 DATA 16,32,127,32,16, 8,28,42,8;8, 8,8,42,28,8, 6
     4,64,64,64,64
```

Program Listing 1. Large Size Screenprinter Subroutine

Once the initialization is complete (one and one-half minutes for the large print and one minute for the small one) this section of the program is not accessed again, regardless of how many different prints are generated by the main program. The rest of the program is then a table look-up followed by printing. The importance of reducing calculations to a minimum becomes clear when you realize that about 26,000 bytes must be sent to the printer to produce one large printout.

The large 7 1/2 × 5 1/2-inch print requires four and a half minutes for the first print, and three minutes for each subsequent one. Comparable times for the small size, 7 1/2 × 4-inch print, as illustrated in Fig. 2, are three and a half and two and a half minutes, respectively.

Using the Screenprinter

Generating prints is easy. The main program must fill two requirements (besides not using overlapping statement numbers): The program must clear 2000 bytes of string storage for exclusive use of the Screenprinter; and variable names beginning with Q should be avoided to prevent conflict with Screenprinter.

Your main program, or driver, generates graphic and alphanumeric characters on the screen. To obtain a printout, the main program issues the command:

GOSUB 10000

The screen, unless modified by an error message, is then reproduced by the printer. The first time the program issues a GOSUB 10000, an initialization routine is called. After the first print, each subsequent GOSUB 10000 causes an immediate printout of the screen.

This initialization is performed only once, provided the driver does not return to a CLEAR or DIM statement, and the user does not type EDIT or RUN between prints (GOTO is all right).

✓ Reader Service—see page 242

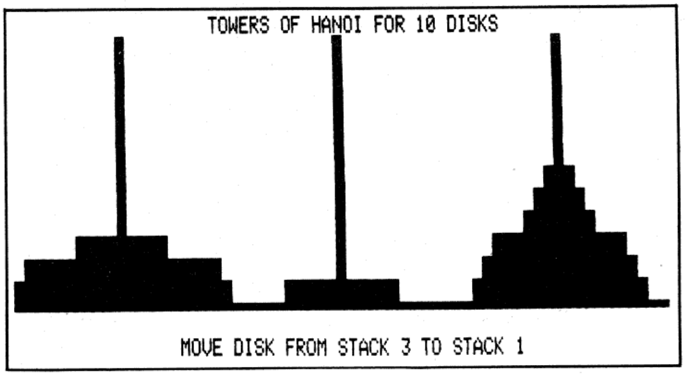


Fig. 2. Example of Small-size Screenprinter Merged with Author's Towers of Hanoi

If you wish to exit while printing, press the B key. This not only stops printing, but also exits the graphics mode. The break key does not reset the printer, hence should not be used once printing has started.

Printing in the Graphics Mode

The printer recognizes certain ASCII characters as control characters. For example, CHR$(3) is interpreted as a signal to enter the graphics mode. Once there, each byte received from the computer is interpreted by the printer as a graphics character. The least significant six bits correspond to a column of six dots, with the least significant bit representing the top dot. The seventh bit should be 0 for contiguous printing, and the eighth, or most significant bit, is ignored by the printer.

The Paper Tiger requires that control characters (such as the vertical tab, CHR$(11), used to end each row of printing) be prefixed with CHR$(3). The control character, CHR$(2), used to exit the graphics mode, must be prefixed with a CHR$(3). The graphic character, CHR$(3), must be prefixed by itself to prevent its being interpreted as a control character.

Because the control characters (CHR$(0) through CHR$(31)) are treated differently by different printer interfaces, and because the TRS-80 has some confusing rules for control characters, Screenprinter avoids them.

This is easily done by adding 128 to all characters before they are sent to the printer. This sets the eighth bit. The extra bit in the eighth position fools the TRS-80 and its printer interface, which think they are sending ordinary characters (greater than ASCII 31) to the printer. But the Paper Tiger ignores the extra bit, the equivalent of subtracting 128 from the transmitted character. The printer thus is able to receive all the control characters as well as alphanumerics and graphics.

Alphanumeric and Graphic Characters

Anyone interested in constructing a software alphabet can adapt the following methods to any printer that generates characters.

Screenprinter reads each alphanumeric character as a five-byte number. Fig. 3(a) shows the letter A imbedded in a 5 × 7 cell. The five columns are numbered


```
10000  ' SMALL-SIZE SCREENPRINTER
10100  FOR QK = 0 TO 2
10550  FOR QQ = 0 TO 7: FOR QK = 0 TO 2 'CREATE GRAPHIC STRINGS
10620  NEXT QJ
10670  DATA ,,, 63,,, ,63,, 63,63,0
10680  DATA ,,63, 63,,63, ,63,63, 63,63,63
```

Program Listing 2. Changes Required in Program Listing 1 to Obtain Small Size Screenprinter

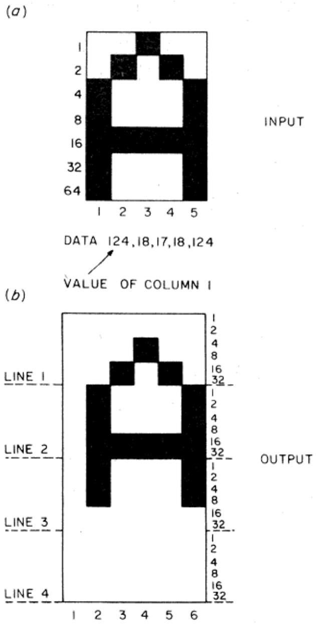


Fig. 3. Example of Software Character Generation

consecutively starting from the left; the seven rows are labeled, starting from the top, with successive powers of two.

The numerical value associated with any column is the sum of the labels corresponding to each black field in that column. The data statement for the letter A consists of five bytes representing the values of the five columns:

DATA 124,18,17,18,124

That's how data is entered for alphanumerics, but is not the way it is sent to the printer. Several problems have to be addressed first.

Firstly, for the contiguous printing necessary to make credible graphics, the Paper Tiger can accept only six, not seven rows. Also, alphanumeric characters on successive lines would touch. Even if we degraded the character set to meet these restrictions, the printout would still be very small.

The solution is to transform the data into large characters that have sufficient white space around them to simulate the TRS-80 screen.

Fig. 3(b) shows the transformed letter A for the large version of Screenprinter. The top six rows correspond to line one on the printout; the next six, to line two, etc. The values are obtained as before, except now there are four values for each column, corresponding to the four lines.

Note that if these values were entered as data, 6 × 4 bytes would have to be read, instead of five bytes. Screenprinter reads only the five bytes, and then, during initialization, uses an algorithm to transform each character into its large counterpart.

Fig. 4(a) shows an example of

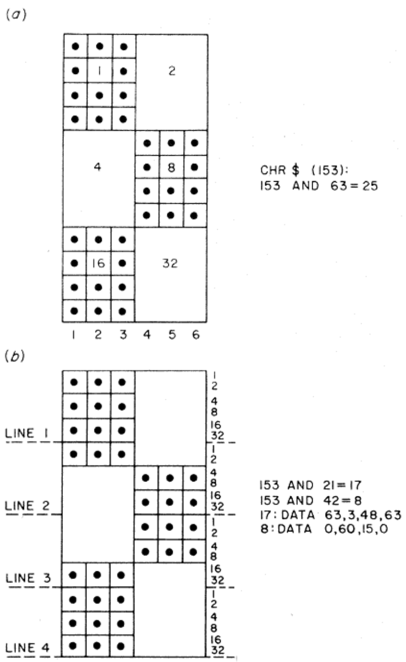


Fig. 4. Example of Software Graphics Generation

Reader Service—see page 242

a TRS-80 graphic character represented by CHR$(153). The blocks are numbered, left to right, starting at the top, with successive powers of two.

Adding the values of the black blocks yields a value of 25. The sum of all blocks is 63. Executing the BASIC logical statement '153 AND 63' yields 25. If any other graphic character had been selected, executing a bitwise AND of its ASCII value and 63 would correspond to the sum of the blocks used to represent that character.

Next, imbed CHR$(153) in a 6 × 24 matrix, as we did for alphanumerics. Using the same method as before, the following values are obtained:

Columns 1-3	Columns 4-6	Line #
63	0	1
3	60	2
48	15	3
63	0	4

The columns on the left correspond to blocks 1 plus 16 (for convenience, call this combination block 17); those on the right, to block 8. The values for block 8 are the same as those for block 4 to its left. Thus specifying the values of blocks 0, 1, 4, 5, 16, 17, 20, 21 is more than enough to determine the value of any combination of blocks. A subset of these blocks is also sufficient, but requires additional calculations. Given the above data, the value corresponding to any graphic character is determined by two bitwise ANDs. Using our previous example:

```
153 AND 21 = 17
153 AND 42 =  8
```

```
10 'DRIVER FOR SCREEN-
   PRINTER
20 CLEAR 2000
30 DEFINT I
40 FOR I=0 TO 1023
50 POKE 15360+I,32 +I-160*
   INT(I/160) 'FILL SCREEN
60 NEXT I
70 GOSUB 10000 'MAKE A
   PRINT
80 END
90 '
```

Program Listing 3. Example of Simple Driver Program, which together with Program Listing 1 produces Fig. 5.

The leftmost three columns are printed by looking up the value associated with block 17, and the rightmost by using the value associated with block 8. Similarly, each ASCII graphic character is masked with 21 and 42 to obtain the proper blocks. The data corresponding to blocks adding to 17 and 8 is read via:

```
DATA 63,3,48,63
DATA 0,60,15,0
```

If you want a three-line, or small-size printout, then line four is dropped for alphanumerics and different data statements must be substituted for the graphic characters.

Different alphanumerics (e.g., a different font) can be created by redefining the 5 × 7 cell of Fig. 3(a) and changing the corresponding data statements.

The Graphic Frame

Variables Q4$ and Q5$ are the upper left and right corners of the frame; QU$ and QB$ are the upper and bottom borders; QL$ represents the left and right sides of the frame; Q8 is the TRS-80 line width.

QT$ is the control character for a vertical tab; QZ$ generates a graphic zero, or blank; and QF$ is a form feed. (You may want to substitute a vertical tab to save paper.)

Line 10560 generates the graphic strings, and the three lines starting with 10590 implement the transformation algorithm for generating alphanumeric strings.

After initialization, the routine issues a form feed, and then enters graphics mode. The border is started, then a line of 64 characters is read from the video memory. Each character is then tested. If alphanumeric, a blank is printed in the first column for that character, and a table look-up provides the next five columns.

If the character is graphic, a table look-up is used for the first three columns, then another one for the next three. Line 10110 provides an escape from printing via the B key.

For each of the 16 lines of text, four lines are generated on the printer for the large Screenprinter and three for the small-sized version.

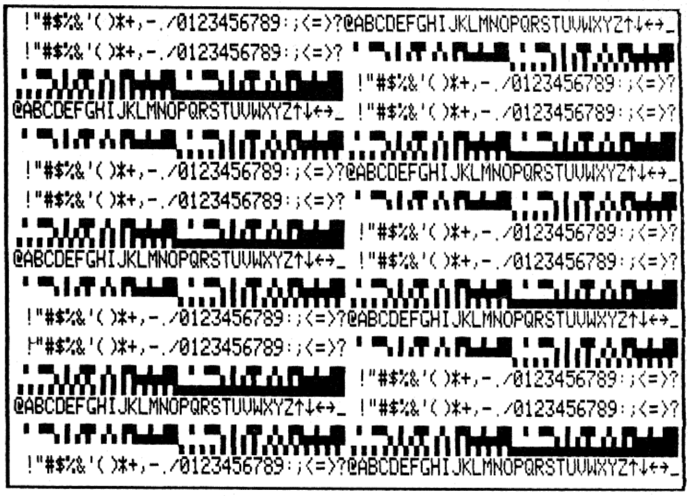


Fig. 5. Example of Screenprinter Output Showing all Characters and TRS-80 Graphics as Obtained via Program in Listings 3 and 1

When the printing is finished, the border is completed. The routine exits from graphics mode before returning to the calling program.

Program Listing 3 is a driver that fills the screen with all possible characters in sequence, as illustrated in Fig. 5.

The three-line Screenprinter was written to offer a smaller print with shorter printing time. It differs from the original in five statements, because three lines, not four are generated for each line on the screen, and because the graphic input data is affected. ■

✓Reader Service—see page 242

Crack the secret of the CLP and take the effort out of editing.

Auto Edit

Dan Rollins
370 N. Cerritos #15-A
Azusa, CA 91720

It's 1 a.m. and Ultra Pong Version 3.2 is finally up and running. You'd suddenly like to speed up execution and save some bytes by deleting spaces and REM statements from the text. You could put off the project; line-by-line editing can be very tiresome. You'd be up till dawn, listing, noting line numbers, editing, deleting, listing some more. Anyway it *does* work (yawn) as it is, after all, so (yawn)...

(Short flourish of trumpets) Wake up! ENTER (stage left) AUTO-EDIT!

Surely you've tried:

```
FOR X = 10 TO 1000: EDIT X :NEXT
    —Forget it
EDIT. + 10  —Wishful thinking
EDIT 10-1000  —Respectable try
EDIT-AUTO ; AUTOEDIT ;EDIT.AUTO
    —No such animal!
```

and then gone back to manual labor.

I didn't give up so easily. I was curious about that "." in LIST., DELETE., and EDIT.. I'll call it the Current Line Pointer (CLP).

That slippery little devil is constantly changing. It points to the last line to be listed, edited, or entered into a program. An ERROR during a program RUN will also reset the CLP, saving many a keystroke in de-bugging. Curiously, ROM assigns the value 0 to "." when used in contexts such as GOTO.,PRINT.,or in mathematical expressions.

Since the CLP changes, I made a few assumptions. CLP must be saved in READ/WRITE memory. It is probably somewhere between the end of VIDEO memory and the start of program text.

Finally, in order for the CLP to point to line numbers greater than 255, CLP must be a two-byte word.

Thus, a short program grew on my screen:

```
10 INPUT "LINE # TO LOCATE"; N
20 MSB = INT(N/256):LSB = N –
   MSB*256
30 FOR X = 16383 TO 17129
40 IF PEEK(X) = LSB IF PEEK(X + 1) =
   MSB,PRINT"FOUND AT";X:STOP
50 NEXT X
```

I set the CLP to a known value by EDIT 20 (and EDIT.) and ran the program, inputting 20. Then I crossed my fingers.

My sophisticated $1.98 monitoring system (transistor radio) buzzed briefly, burped, and my display then announced a single message—FOUND AT 16620. Faster than you can say "Indiscriminate POKEing in low memory will turn your computer into smoldering slag", I typed POKE 16620,30 (slight hesitation) and ENTER. The satisfying response to LIST. tells the story:

```
30 FOR X = 16383 TO 17129
```

This was proof positive that the " . " could be used as a variable to indicate line numbers. I had the single clue I needed. The rest of AUTO-EDIT 1 seemed to just click into place.

Two Styles

AUTO-EDIT comes in two styles: BASIC and assembly language. The BASIC version may be typed into an existing program.

Begin with an EDIT##. Thereafter, four keystrokes: RUN ENTER will execute EDIT for the next line number.

For readers with a sweet tooth, I recommend the assembly language version. After initializing the CLP (EDIT##), a new command SHIFT ENTER produces EDIT mode for successive program lines.

This capability, which Microsoft somehow neglected in their Level II package, is valuable at all stages of program development. Repeated use of SHIFT ENTER is much like the single-step LIST of LEVEL I. During program clean-up, AUTO-EDIT is priceless.

Both versions of the program go through the same series of steps:

1) Getting the value of CLP;
2) Finding the number of the following program line;
3) Updating the CLP to this value;
4) Executing EDIT for that line.

AUTO-EDIT BASIC saves the " . " in the variable named CLP. Lines two to four form a loop. Starting at the beginning of your BASIC program, it leapfrogs from line to line, checking the line number (bytes three and four of any line) against CLP. The loop is broken when CLP is exceeded. The " . " is given the new value in line three. Line five leaves BASIC for the mysteries of EDIT mode.

AUTO-EDIT 2 (assembly language) requires a few preliminaries. The first order of business is the setting up of a patch from BASIC. ROM occasionally CALLs certain addresses in RAM, known as exits, asking for more instructions. This is to maintain upward compatibility with DOS and allow just such extensions to BASIC as AUTO-EDIT. Usually, all that is found

there is a RETurn directive. There is one such exit which is called every time ENTER is hit.

The first 12 bytes of the program replace this return with instructions to jump to 7FE9H and resume execution there. The next byte, a HALT instruction, stops the CPU long enough to allow a hardware interrupt to take place. This results in handing over control to BASIC, much the same as pressing the reset button. Your computer sees this part of the program only once, when you type /32733 after the SYSTEM prompt (*?).

Now, whenever you hit ENTER, the Z-80 microprocessor receives the instructions starting at 7FE9H. Here it is told to see if the SHIFT key is down. If it's not, the RETURN instruction tells it to quit wasting its time and get down to the business at hand! However, if SHIFT *is* being pressed, it is time to then execute AUTO-EDIT.

The CLP is saved in the DE register pair. The hard part is locating the next higher line number, and it is taken care of by FNDLIN, a subroutine located in ROM. BASIC calls FNDLIN to determine where to take up execution after a GOTO or GOSUB branching directive. This byte saver uses a leapfrog technique similar in principle to lines two and three of AUTO-EDIT 1.

When the line number in DE is found, BC returns, pointing to the start of that BASIC line, and HL holds the address of the following one. Since the jump to EDITOR requires the line number to be in HL, this seems ideal. Unfortunately, there is a special case which causes an inaccuracy.

If, while using AUTO-EDIT, you came across a REMark or other unwanted line, you might choose to delete it with Hack (H), 99Delete (D), Kill (K) SHIFT ENTER or, from DIRECT mode, DELETE.. When this happens, the CLP will point to a nonexistent line. No match will be found, so FNDLIN will return BC and HL, pointing a notch higher than expected. This will result in AUTO-EDIT skipping a line.

```
0'                    * AUTO-EDIT 1 *
  * TYPE THESE LINES INTO AN EXISTING PROGRAM.       *
  * INITIALIZE WITH EDIT ##. REPEATEDLY RUN TO        *
  * PRODUCE EDIT MODE FOR SUCCESSIVE LINES.          *

1 CLP = PEEK(16620) + PEEK(16621)*256 :X = 17129
2 LN = PEEK(X + 2) + PEEK(X + 3)*256
3 X = PEEK(X) + PEEK(X + 1)*256 :IF LN <= CLP GOTO 2
4 POKE 16620,PEEK(X + 2) :POKE 16621,PEEK(X + 3)
5 EDIT.
```

Program Listing 1. AUTO-EDIT 1

This possibility could be checked using the CPU flags. However, I chose to generalize the result without using flags, making the program shorter. The DE register pair is incremented *before* the CALL to FNDLIN, so it never points to CLP, but never higher than the next possible line number. Instead of HL, BC is used as the reference in determining the new value for the CLP.

Now BC is loaded into HL which is bumped twice to point to the bytes containing the target line number. We want the *value* of these bytes, not their location, so a call is made to a routine which loads HL with the contents of the address pointed to by HL: LD HL,(HL) : the forgotten Op Code!

Finally, the jump to the EDITOR updates the " . " and produces EDIT mode.

Only one question remains. What happens when there are no more lines to EDIT? Does the program leapfrog its merry way into a black hole? What are these wisps of acrid blue smoke rising from my keyboard? I'll sue!

Put your mind at ease. Trust me; I wouldn't even melt your ice cream. . . let alone be responsible for the Three Mile Island of microcomputing!

After the final EDIT, SHIFT ENTER produces predictable, and not unpleasant effects. If the last line of Ultra Pong is numbered less than 256, AUTO-EDIT will point to this line until reinitialized. Otherwise, the message UL ERROR is displayed and *zap*! you're back in EDIT for the *first* program line!

Frankly, I'm not sure why this happens. Level II is a lengthy program and T-BUG balks at breakpointing ROM. My theory is that during error processing, CLP is set to FFFFH and a CALL is made to the patch at 41AFH. Anyway, no harm is done and it is an interesting puzzle. If you can trace the exact sequence, I'd like to hear from you.

Applications

The time is 1:15 a.m. Spaces and REMarks have been removed and Ultra Pong Version 3.3 is operational at a somewhat faster speed. PRINT MEM returns a larger value. You are about to take the transcendental step to Super Ultra Pong Version 4.0!

Ready? Step through your program, using AUTO-EDIT, of course, making note of the variables most often used. Pay attention to those within loops. Also, jot down the values of constants often accessed. As a suggestion, the first place to start

```
0'* THIS PROGRAM POKES THE MACHINE CODE          *
  * AND SETS THE PATCH FOR AUTO-EDIT 2           *

10 X = 32745                                  '* top of 16K
20 READ Y : IF Y = -1 GOTO 50
30 POKE X,Y :X = X + 1 :C = C + Y
40 GOTO 20
50 IF C<>2534 PRINT "BAD CHECKSUM" :STOP
60 POKE 16815,195 :POKE 16816,233 :POKE 16817,127
90 DATA 58, 128, 56, 31, 208, 237, 91, 236, 64, 19, 205, 44
100 DATA 27, 197, 225, 35, 35, 205, 63, 27, 195, 102, 46, -1
```

Program Listing 2. AUTO-EDIT 2

HARDWARE BREAKTHROUGH

DESIGN SOLUTION INC. presents the AN-SERIES DEVELOPMENT SYSTEM

DIGITAL SPEECH PROCESSOR

Experiment with digital speech synthesis and recognition. using a **BASIC** editor program provided with this system. Speech is entered, digitized and stored in memory or in disk files. Vocabulary files are construced using digital core images of your voice. Labeled and indexed sounds, words and phrases are then available for use in your **BASIC** programs. Different vocabulary files can be loaded for different **BASIC** Programs, etc. while the digital speech processor will operate on any model I, Level II **TRS-80*** system. We recommend a level II with disk drive to reduce voice image loading times. Experiment with data compression from **BASIC**, by increasing and decreasing sample rates. Develop your own voice recognition programs. Simulate low pass, band pass, and high pass filters, etc. This is truly a software man's dream-come-true. Enter a string of numeric data and let your **TRS-80** repeat it. Let your **TRS-80** dictate data files for error checking. The sky and your imagination are the absolute limit!

2716-2732 EPROM PROGRAMMER

At last! An affordable **EPROM** programmer for your **TRS-80***. Program either 2716 or 2732 **EPROMS** from a **BASIC** program in less than 300 seconds. With **DESIGN SOLUTION'S SOFTWARE** you are able to program **(BURN IN) EPROM** from **TRS-80** memory, **VERIFY** data transfer, copy **ROM** to **EPROM**, or load **TRS-80** memory from **ROM** or **EPROM**. **THE AN-551** is a must for all small system development packages. Unit comes complete with external power supply and operation manual. The 40-pin edge card connector is **GOLD** plated for extra long life. **ZERO INSERTION FORCE SOCKETS** simply plug into the front panel socket of the **AN-551** if quick changes are required. (ZIF socket not included.)

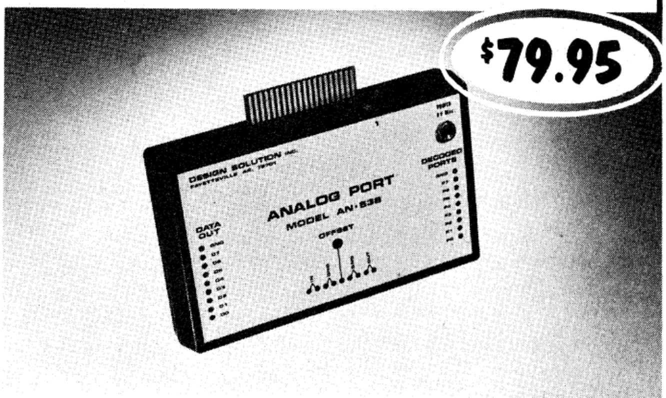
DIGITAL PORT INTERFACE

Now you can **BREADBOARD** your digital projects with ease. The **AN-511 DIGITAL PORT** provides all the hardware necessary to interface your digital projects with the **TRS-80***. From **MACHINE LANGUAGE** or **BASIC** programs you can now access and control:

- **8 BITS OF OUTPUT DATA FROM TRS-80**
- **8 BITS OF INPUT DATA TO THE TRS-80**
- **8 DECODED PORT ADDRESS (DEVICE CONTROL) LINES**
- **8 BIT DATA BUS FROM Z-80 IN TRS-80**

Using the **IN** and **OUT** commands in **BASIC** or their machine language counter-parts, 8 bit data values can be moved from external hardware projects into the **TRS-80** and vice-versa. With the 8 decoded port outputs up to 8 additional bytes of input or output data can be accessed. The **AN-511** is supplied with external power supply and complete operation and projects manual.

ANALOG PORT INTERFACE



The **SUPER BOX!** The **ANALOG PORT, MODEL AN-538**, is a very versatile **ANALOG TO DIGITAL AND DIGITAL TO ANALOG CONVERTER**. All conversion operations are under **TRS-80*** program control. Experiment with **MUSIC** and **VOICE SYNTHESIS**. From machine language or **BASIC** programs you can manipulate analog signals, digitize voltages and measure them with the **TRS-80**. Generate various waveforms, noise, speech patterns, etc., and output them as analog signals. The **AN-538** also provides a mini-digital port. The 8 bit output data to the D/A converter is accessable through the front panel connector, along with 8 extra decoded port (device control) lines. All conections are made using standard #22 solid wire. Lines push through the front panel, received underneath by special connectors, ensuring positive connection. The **AN-538** comes complete with external power supply and operation-project manual.

All connections made to **AN-SERIES** products from your breadboard are simply pushed through the front panel. Custom connectors on the P.C.B. provide super reliable connection for thousands of operations. All **AN-SERIES** products are warrented for a full **90-DAYS** under **DSI's** limited warrenty policy. Complete documentation is provided for each model in an attractive folder, including theory of operation, and special interest projects and applications. All units are supplied with external power supply modules that connect through a mini-jack on the front panel. A 40-conductor cable is required for connection with your **TRS-80** and is **NOT SUPPLIED** by **DESIGN SOLUTION**, but is available from your local **RADIO SHACK STORE**. (Components are 1(278-771) 40 conductor ribbon cable and 2(276-1558) edge card connectors.)

***TRS-80** is a registered trademark of **RADIO SHACK, a TANDY COMPANY**.

DESIGN SOLUTION INC. ✓123
BOX 1225, FAYETTEVILLE, AR 72701

C.O.D. add $3.00 — Other add $1.50 S/H.
Orders and Information: (501) 521-0281

looking is at your cursor-control routines, where dots and lines are manipulated around the screen.

Now, create a line early in the program. Use DEFINT var, var, var, etc. for any of these variables which will remain integers throughout a run. Add a list of *new* variables, equating them to the constants noted earlier. Your line might look like this:

```
10 DEFINT A - C,I - L,X,Y:I0 = 0:
   I1 = 1:IU = -64 :etc. . .
   :UP$ = CHR$(91):DN$ =
   CHR$(10) :etc. . .
```

Finally, review the listing, EDITing the occurrence of each defined constant to its variable equivalent.

Presto! You have just accelerated execution by as much as 20 percent! Perhaps a proof is in order for you skeptics.

Time the delay caused by each of the following lines:

```
>FORX = 1TO5000:Z = 1 + 2:NEXT
>Y1 = 1:Y2 = 2:FORX = 1TO5000:
   Z = Y1 + Y2:NEXT
>DEFINTX - Z:Y1 = 1:Y2 = 2:FORX
   = 1TO5000:Z = Y1 + Y2:NEXT
>FORX = 1TO5000:
```

All three do the same thing, but I got about a three percent speed-up by replacing the constants, and an overall savings of 25 percent with integer variables.

A drawback is that a line like:

```
1010 IF A$ = CHR$(10) THEN
     Y = Y - 1 : RETURN
```

may be easier to debug than the more efficient:

```
1010 IFA$ = DN$Y = Y + DN:RETURN
```

so I suggest saving this modification until after debugging is completed.

```
                    ;
7FDD                           ORG    7FDDH             ;  = 32733  DECIMAL
41AF                    PATCH  EQU    41AFH             ;  = 16815  DECIMAL
2E66                    EDITOR EQU    2E66H
4ØEC                    CLP    EQU    4ØECH
1B2C                    FNDLIN EQU    1B2CH
7FE9                    ATOEDT EQU    7FE9H
                    ;
                    ;**********************************************************
                    ;**     THIS PART WILL SET UP A  "PATCH"  FROM          **
                    ;**    BASIC.  41AFH IS CALLED WHENEVER <ENTER>          **
                    ;**    IS HIT.  REPLACE THE  RETURN  DIRECTIVE           **
                    ;**             THERE, WITH  A JUMP TO                   **
                    ;**                    AUTO-EDIT                         **
                    ;**********************************************************
                    ;
7FDD 2EC3                      LD     L,ØC3H            ;C3 MEANS "JUMP TO"
7FDF 22AF41                    LD     (PATCH),HL        ;  WRITE IT IN MEMORY
7FE2 21E97F                    LD     HL,ATOEDT         ;ATOEDT IS LOCATION
7FE5 22BØ41                    LD     (PATCH+1),HL      ;TO JUMP.   WRITE IT
7FE8 76                        HALT                     ;RETURN TO ROM CONTROL
                    ;
                    ;==========================================================
                    ;==                      AUTO-EDIT                       ==
                    ;==       INITIALIZE THE PROGRAM WITH   EDIT ##          ==
                    ;==    THEREAFTER, HIT <ENTER> WITH THE   <SHIFT>        ==
                    ;==    KEY PRESSED TO EDIT NEXT PROGRAM LINE.            ==
                    ;==========================================================
                    ;
7FE9 3A8Ø38             ATOEDT LD     A,(388ØH)         ;CHECK FOR <SHIFT>
7FEC 1F                        RRA                      ;BY ROTATING IT'S BIT
                                                        ;INTO THE CARRY FLAG
7FED DØ                        RET    NC                ;NOT PRESSED? GOTO BASIC
                                                        ;ELSE..... AUTO-EDIT!
                    ;
7FEE ED5BEC4Ø                  LD     DE,(CLP)          ;GET THE  " . "
7FF2 13                        INC    DE                ;CHECK FOR NEXT HIGHEST
7FF3 CD2C1B                    CALL   FNDLIN            ;WITH THIS ROM ROUTINE
                                                        ;  WHICH RETURNS BC WITH
                                                        ;    ADDRESS OF NEXT LINE

7FF6 C5                        PUSH   BC                ;PUT ADDR INTO HL
7FF7 E1                        POP    HL                ;  AND BUMP IT TO
7FF8 23                        INC    HL                ;    POINT TO
7FF9 23                        INC    HL                ;       LINE NUMBER
7FFA CD3F1B                    CALL   1B3FH             ;ROUTINE TO LD    HL,(HL)
                                                        ; HL  HAS NUMBER
                                                        ;  OF LINE TO  EDIT
7FFD C3662E                    JP     EDITOR            ;    EXECUTE  EDIT
ØØØØ                           END
ØØØØØ TOTAL ERRORS
```

Program Listing 3. Assembled Source Code for AUTO-EDIT 2

Etc., Etc.

Other uses include the following:

- Fix GOTOs and GOSUBs after using BASIC BASIC RENUMBERER (see *80 Microcomputing*, Jan/80).
- Change PRINT statements to LPRINT.
- Alter variable names in appended subroutines to avoid conflicts with the main program.
- Prepare program listings for the line printer. Lengthy lines may be broken up from EDIT mode by placing the printer command, "line feed with carriage return" at convenient spots along the line. Just type C ENTER over a space!
- This little-known trick makes possible the Pauper's Pencil word processor:

```
10 A$(1) = "              "
20 A$(2) = "              "
30 A$(3) = "              "
300 FOR X = 1 TO 3: LPRINT A$(X); : NEXT
```

Simply Insert text between the quotes. When finished, justify margins by inserting spaces and line feeds as needed. Multiple line feeds must have a space between each, and block moves are done by changing (C) the subscripts to A$.

Since AUTO-EDIT keeps us in EDIT mode, Pauper's Pencil becomes practical, and almost easy.

Use T-BUG to code AUTO-EDIT 2 and save it with the punch (P) command: P 7FDD 7FFF ATOEDT. The straight-line logic of the program makes it relocatable if the bytes at 7FE3-4 are altered to reflect the change. To code at the top of 4K machines, start modifying (M) at 4FDDH. Change only the byte at 4FE4H from 7F to 4F.

Disk spinners—watch out for that HALT instruction. It tends to re-boot your DOS. A jump to 1A19H works fine instead.

Though AUTO-EDIT is useful enough to be a stand-alone program, I always load it in with QMERGE (a program which appends BASIC subroutines without wiping out the base program) and T-SHORT (a shorthand for BASIC commands from Web Associates), which includes a programmable key. I won't code a line without my BASIC extension kit in memory.

If you must code the program from BASIC, Program Listing 2 POKEs the decimal values of the machine code and sets the patch.

Oh yes, MEMORY SIZE should be set at 32732 to avoid having ROM push stack patch data all over program code.

Assembly language programmers, now that you know where EDITOR is, can you use a subroutine which will search (S) a 255-byte buffer for a particular value? How about one to insert (I) a byte and move everything up one address? Have fun.■

Now NRI takes you inside the world's most popular microcomputer to train you at home as the new breed of computer specialist!

NRI teams up with Radio Shack to teach you how to use, program and service microcomputers...make you the complete technician.

It's no longer enough to be just a programmer or a technician. With microcomputers moving into the fabric of our lives (over 200,000 of the TRS-80™ alone have been sold), interdisciplinary skills are demanded. And NRI can prepare you with the first course of its kind, covering the complete world of the microcomputer.

Learn At Home in Your Spare Time

With NRI training, the programmer gains practical knowledge of hardware, enabling him to design simpler, more effective programs. And, with advanced programming skills, the technician can test and debug systems quickly and easily.

Only NRI gives you both kinds of training with the convenience of home study. No classroom pressures, no night school, no gasoline wasted. You learn at your convenience, at your own pace. Yet you're always backed by the NRI staff and your instructor, answering questions, giving you guidance, and helping you over the tough spots.

Training includes TRS-80 computer, transistorized volt-ohm meter, digital frequency counter, and the NRI Discovery Lab with hundreds of tests and experiments.

(TRS-80 is a trademark of the Radio Shack division of Tandy Corp.)

Explore the TRS-80 Inside and Out

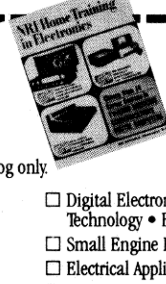
NRI training is hands-on training, with practical experiments and demonstrations as the very foundation of your knowledge. You don't just program your computer, you introduce and correct faults ...watch how circuits interact...interface with other systems...gain a real insight into its nature.

You also build test instruments and the NRI Discovery Lab, performing over 60 separate experiments in the process. You learn how your trouble-shooting tools work, and gain greater understanding of the information they give you. Both microcomputer and equipment come as part of your training for you to use and keep.

Send for Free Catalog... No Salesman Will Call

Get all the details on this exciting course in NRI's free, 100-page catalog. It shows all equipment, lesson outlines, and facts on other electronics courses such as Complete Communications with CB, TV and Audio, Digital Electronics, and more. Send today, no salesman will ever bother you. Keep up with the latest technology as you learn on the world's most popular computer. If coupon has been used, write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016
NO SALESMAN WILL CALL
Please check for one free catalog only.



All career courses approved under GI Bill.
☐ Check for details.

- ☐ Computer Electronics Including Microcomputers
- ☐ TV/Audio/Video Systems Servicing
- ☐ Complete Communications Electronics with CB • FCC Licenses • Aircraft, Mobile, Marine Electronics
- ☐ CB Specialists Course
- ☐ Digital Electronics • Electronic Technology • Basic Electronics
- ☐ Small Engine Repair
- ☐ Electrical Appliance Servicing
- ☐ Automotive Mechanics
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Refrigeration, & Heating including Solar Technology

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council 179-021

Reader Service—see page 242

PMC-80

Level II 16K at $645

SOFTWARE COMPATIBLE

- Reads all Level II BASIC tapes
- Reads all SYSTEM tapes
- Full range of peripherals
- Video output for monitor and TV
- Optional FASTLOAD at 8000 baud
- Optional Upper/Lower case

The PMC-80 is a "work-alike" computer to the popular TRS-80* Model I, Level II by Tandy, Radio Shack. The PMC-80 has 16K bytes of RAM and the complete Level II 12K BASIC ROM by Microsoft that makes it 100% software compatible with programs from Radio Shack and from the hundreds of other independent suppliers. The built-in cassette player reads standard Radio Shack programs for the TRS-80.*

Sold through computer stores.

The PMC-80 will operate with any of the many peripherals Radio Shack and other independent vendors have invented to plug into the TRS-80.* Most importantly, the Interface Adapter permits Expansion Interfaces with memory expansion to 48K to be added. An Expansion Interface will also permit the addition of Radio Shack compatible 5¼″ disks and disk operating systems, RS 232, printers, etc.

*TRS-80 is a registered trademark of Tandy, Radio Shack.

Personal Micro Computers, Inc. ✓422

475 Ellis Street, Mountain View, CA 94043 (415) 962-0220

Fastload

FOR TRS-80* MODEL I USERS ONLY

16 Times Normal Speed

*TRS-80 is a trademark of Tandy Corp.

- High speed load TRS-80* Level II cassettes
- Input 15K byte Level II program in 15 seconds
- Search BASIC or SYSTEM programs by name

Unlike other high speed tape input devices, FASTLOAD uses standard format cassettes. Therefore, there is no need to re-record on other media. At 8000 baud, FASTLOAD is faster than disk for short programs. FASTLOAD reads tapes at the fast-forward speed of the CTR-41 cassette recorder. The recorder can also be used for CSAVE at the normal speed.

FASTLOAD connects to the 40 pin I/O or to the Expansion box. The control program does not use computer memory because it is in a built-in PROM. Other valuable features are keyboard debounce program, automatic key repeat routine and key-beep via cassette speaker. Price is $188.00 for FASTLOAD and $95.00 for the modified CTR-41 recorder.

✓422

Personal Micro Computers Inc.

475 Ellis Street, Mountain View, CA 94043 (415) 968-1604

This analysis program, based on Lincoln-Peterson techniques, even checks its own accuracy.

Population Estimation

Dennis R. Solomon
4215 Grand Ave.
Des Moines, IA 50312

Before the computer began replacing pretty secretaries in front offices, the science world used them to replace their worn out slide rules. Today the word computer stirs up an image of the machine that keeps the books, prints the checks and produces all those bills. Scientific studies are still suited to microcomputers, however, and have useful applications.

Since the environment is a popular topic, ecology is a science that often interests everybody. The numerical tangle of statistics is one area where the micro can prove its worth as an instrument of analysis.

Ecology deals with natural populations. Before studying a population in depth it is often necessary to obtain information on its size. Most studies use one of several techniques to estimate size. The Lincoln-Peterson Index is one of these techniques.

Let's say we want to estimate

$$\text{Population size} = \frac{\begin{pmatrix}\text{number of marked}\\ \text{animals released}\end{pmatrix} \times \begin{pmatrix}\text{number of animals}\\ \text{captured in second}\\ \text{sample}\end{pmatrix}}{(\text{number of marked animals in second sample})}$$

Formula 1.

$$\text{Population size} = \frac{50 \times 100}{24} = 208 \text{ groundhogs}$$

Example 1.

Program Listing

```
10 'POPULATION STATISTICS:  LINCOLN-PETERSON INDEX

20 'WRITTEN BY:  DENNIS SOLOMON/2824 GRAND AV/DES MOINE
   S, IOWA                          50312/515-280-8174
30 CLS:PRINTTAB(16);"*** LINCOLN-PETERSON INDEX ***":PR
   INT:PRINT:PRINT
40 PRINT"THE LINCOLN-PETERSON INDEX IS USED IN ECOLOGIC
   AL RESEARCH"
50 PRINT"TO ESTIMATE THE NUMERICAL SIZE OF A NATURAL PO
   PULATION.":PRINT
60 PRINT"TO USE THIS METHOD OF CALCULATING POPULATION S
   IZE A "
70 PRINT"KNOWN NUMBER OF MARKED ANIMALS ARE RELEASED IN
   TO THE"
80 PRINT"POPULATION, AND AFTER A PERIOD OF TIME A NUMBE
   R OF"
90 PRINT"MEMBERS OF THE POPULATION ARE CAPTURED.  BY CO
   MPARING"
100 PRINT"THE NUMBER OF MARKED VS UNMARKED ANIMALS, THE
    SIZE OF"
110 PRINT"THE POPULATION CAN BE ESTIMATED.":PRINT
120 PRINT"PRESS ANY KEY TO CONTINUE..."
130 IN$=INKEY$:IF IN$=""THEN130  ELSECLS
140 PRINT"ENTER THE NUMBER OF MARKED ANIMALS INTRODUCED
    INTO"
150 PRINT"THE POPULATION:   ";:INPUTNM:PRINT
160 PRINT"ENTER THE TOTAL NUMBER OF ANIMALS IN SAMPLE:
    ";:INPUTT:PRINT
170 PRINT"ENTER THE NUMBER OF MARKED ANIMALS IN SAMPLE:
     ";:INPUTP:PRINT
180 UP=ABS(T-P)
190 N=(NM*T)/P
200 CP=P/T:CQ=(T-P)/T
210 C1=CP+(1.96*(SQR((CP*CQ)/T))):C1=ABS(NM/C1)
220 C2=CP-(1.96*(SQR((CP*CQ)/T))):C2=ABS(NM/C2)
230 CLS:PRINT"LINCOLN-PETERSON INDEX:    ";N:PRINT
240 PRINT"LOWER CONFIDENCE LIMIT:    ";C1:PRINT
250 PRINT"UPPER CONFIDENCE LIMIT:    ";C2:PRINT:PRINT:P
    RINT:PRINT
260 PRINT" THIS MEANS THAT THE POPULATION IS ESTIMATED
    AT";N;"."
270 PRINT"THERE IS A 95% CHANCE THAT THE ACTUAL SIZE OF
    THE";
280 PRINT" POPULATION":PRINT"WILL BE GREATER THAN";C1;"
    AND ";
290 PRINT"LESS THAN";C2;"."
```

the number of groundhogs chewing up crops in the lower forty. First we capture some of the critters and mark them with dye. They are then returned to the area under study and released. Several days are allowed for the marked animals to randomly distribute themselves in their environment before a second group is captured. We are concerned with the number of marked animals and the number of unmarked animals found in the second sample; with this information, we can calculate the number of groundhogs in the fields using Formula 1.

If we originally captured 50 groundhogs, marked them with dye and turned them loose, and later captured 100 groundhogs and found 24 of them marked we would have the equation in Example 1.

We now know we are dealing with a population of about 208 groundhogs. Before we move on to more studies, we'll need to know how accurate our sample is.

$$\text{Upper limit} = \frac{\text{(number of marked animals)}}{p - 1.96 * \text{SQR((p*q)/size of second sample)}}$$

$$\text{Lower limit} = \frac{\text{(number of marked animals)}}{p + 1.96 * \text{SQR((p*q)/size of second sample)}}$$

$$\text{where } p = \frac{\text{number marked animals in second sample}}{\text{size of second sample}}$$

$$\text{and } q = \frac{\text{number unmarked animals in second sample}}{\text{size of second sample}}$$

Formula 2.

Estimating Accuracy

Accuracy is determined by calculating the confidence limits. This is a range of numbers in which we can be 95 percent sure that the actual size of the population occurs.

This is calculated with Formula 2. This provides us with an upper limit of 320 groundhogs and a lower limit of 54.

The Program Listing shows these statistics in a BASIC program written on a Level II TRS-80. It will run under most BASICs by deleting the CLS command in line 30 and deleting line 130.

Lines 190 through 220 contain the formulas usable on any system. The population size is calculated in 190. Next, the decimal fractions p and q are calculated in line 200. Finally, the upper and lower limits of the estimate are calculated in lines 210 and 220. ■

Variable	Program Use
NM	Number of animals captured, marked, then released. (First sample size)
T	Size of second sample
P	Number of marked animals in second sample
N	Lincoln-Peterson Index (estimate of population size)
CP	Decimal fraction of marked animals in second sample (p)
CQ	Decimal fraction of unmarked animals in 2nd sample (q)
C1	Lower confidence limit
C2	Upper confidence limit

Table 1. Variables

Look before you leap into the abyss of real estate. This program shows you how to evaluate potential investments.

Landlord

James A. Tuohy
5916 Bluebonnet Rd.
Charlotte, NC 28212

How do you get on the profit side of inflation? This is a survival question. There are several paths that lead to the oasis of profits in the desert of higher prices and lower purchasing power. I find owning income producing properties the best method for me.

Real estate is one investment that you have some control over. It is tangible, you can drive out and see it, and one of its big advantages is that Uncle Sam lets you keep some or all of your profits through expense and depreciation deductions.

I turned to my trusty computer to help me analyze various properties available for sale. With this program, I can sit down with a list of available properties and go through them quickly, to weed out the ones that are definitely losers, and work with the ones that look like possible purchases. This program lets you change the variables to come up with the best profit offer. My approach to negotiating is "Owners price, my terms—my price, owners terms".

The Income Property Evaluator program is my screening program, setting up the boundaries of my offers. I always work up several possibilities then start by making an offer for the one that promises to be most profitable and inexpensive, then counter-offer with the next best, etc., until I buy or we reach a point of return that is lower than I require. If the owner insists on an offer that is not profitable for me, I look elsewhere.

You will have to modify this program for your city. Line 260 sets the property taxes; you will have to discover the tax valuation of the property, or use the formula shown, changing the variables. I based the value on the proposed purchase price because some cities re-evaluate property when it is sold. This

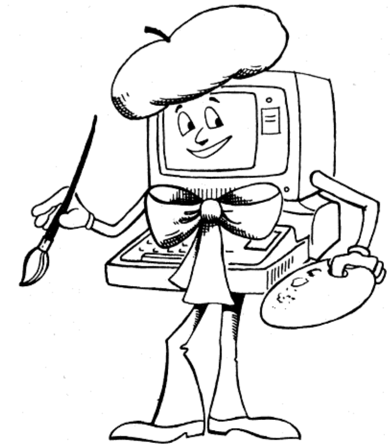
```
1 REM  INCOME PROPERTY EVALUATOR  I
2 REM  BY   JAMES  A.  TUOHY
10 CLS
20 DATA PURCHASE PRICE $,DOWN PAYMENT     %,ADDITIONAL
     CLOSING COSTS $,MONTHLY RENTAL INCOME $
30 DATA VALUE OF LAND (% OF PURCHASE PRICE),TAX BRACKET
      (% OF GROSS INCOME PAID IN TAXES),  PROPERTY INFL
     ATION RATE %
40 DATA YEARS OF 1ST MORTGAGE,INTEREST RATE ON 1ST MORT
     GAGE %
50 B$ ="INCOME  PROPERTY  EVALUATOR  I"
60 PRINT@20,B$
70 FOR I=1 TO 9: READ A$
80 PRINT I;A$;"   ";:INPUT A(I)
90 NEXT I
100 GOTO170
110 RESTORE
120 FOR I=1 TO Q
130 READ A$
140 NEXTI
150 PRINTA$
160 INPUT A(Q)
170 RESTORE
180 CLS:PRINT@20,B$
190 FOR I=1TO9
200 READ A$
210 PRINTI,A$,A(I)
220 NEXTI
230 INPUT"IF ALL ENTRIES ARE CORRECT ENTER #10 , IF NOT
      ENTER THE # TO BE CORRECTED";Q:CLS
240 IFQ>=10 GOTO260
250 IFQ>0 GOTO 110
260 PT=((A(1)*.80)/100)*1.65
270 IC=(A(1)/1000)*8.25
280 L=A(1)-(A(1)*(A(2)/100))
290 S=1+A(9)/1200
300 M=L*(((A(9)/1200)*S[(A(8)*12))/((S[(A(8)*12))-1))
310 ME=A(4)*.15: TE=((A(2)/100)*A(1))+A(3)
320 PRINT"PURCHASE PRICE $ ";A(1);"  TOTAL DOWN PYT $";
     TE;"WHICH IS ";(TE/A(1))*100;" %"
330 PRINT:PRINT"ITEM","ANNUAL AMOUNT"
340 PRINT"MORTAGE PYT",M*12
350 PRINT"PROPERTY TAX",PT
360 PRINT"INSURANCE",IC
370 PRINT"MGT FEE/MAINT.",ME*12
380 PRINT
390 PRINT"RENTAL INCOME",A(4)*12
400 X=(M*12)+PT+IC+(ME*12)
410 PRINT:PRINT"C A S H  F L O W",(A(4)*12)-X
```

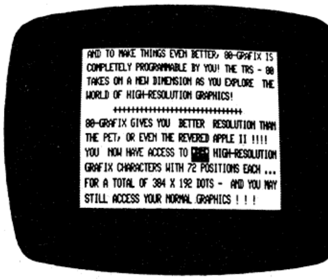
Program continues

FROM **PROGRAMMA**

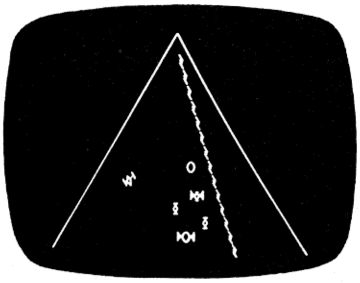
HI-RESOLUTION GRAPHICS FOR THE TRS-80®

LOWER CASE
The 80-GRAFIX board includes two sets of lower case characters at no additional cost.





INVERSE VIDEO
The 80-GRAFIX board allows you to do inverse video to high-light your screen displays.



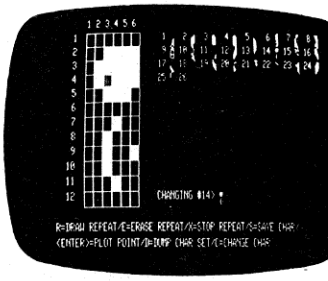
DEMONSTRATION PROGRAMS
The 80-GRAFIX board is supplied with a Character Generator software and several demonstration programs.

FINALLY, AT LAST. . .

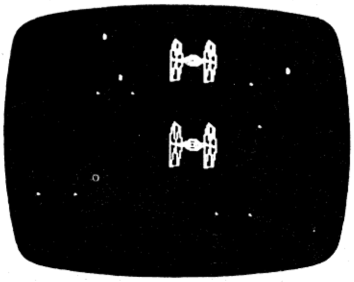
HI- RESOLUTION GRAPHICS is available for your TRS-80 computer system. The 80-GRAFIX board from PROGRAMMA International, Inc. gives your TRS-80 high resolution capability that is greater than the Commodore CBM/PET or even the revered APPLE II.

80-GRAFIX gives the TRS-80 an effective screen of 384X192 pixels, versus the normal 127X192 for the TRS-80, 80X50 for the CBM/PET, or the 280X192 of an APPLE II. As an added feature, 80-GRAFIX offers you lower case characters at no additional cost. Of course, you can also create your own set of up to 64 original characters using the supplied Character Generator software.

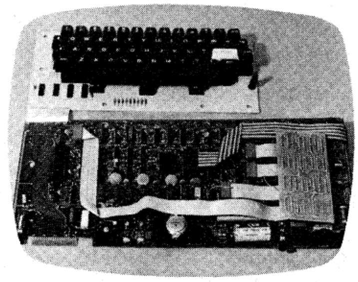
The 80-GRAFIX board is simple to install (note that this voids your Radio Shack warranty), and programming is done through BASIC. 80-GRAFIX opens up a whole new realm of software development and excitement never dreamed of for the TRS-80!



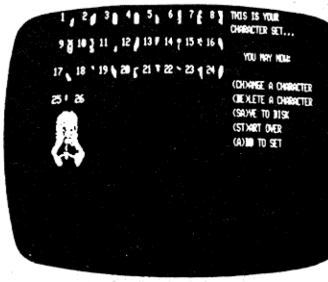
CHARACTER GENERATOR
The supplied character generator software allows you to create your own character set of up to 64 original characters.



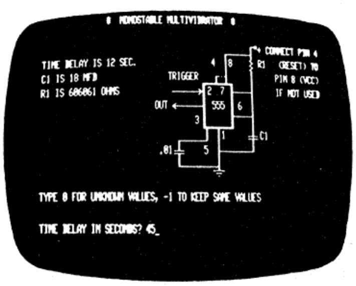
REAL-TIME GRAPHIC GAMES
With the 80-GRAFIX board you can write exciting real-time games using BASIC.



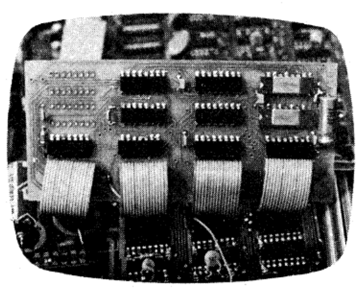
EASY INSTALLATION
The 80-GRAFIX board is simple to install and fits inside the TRS-80 case.



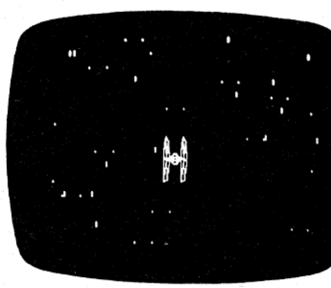
GRAPHICS GALORE
The 80-GRAFIX board and the supplied Character Generator allow you to become an artist.



ELECTRONIC DESIGN
The 80-GRAFIX board has unlimited application in Electronic design and Education.



80-GRAFIX HI-RESOLUTION
Finally, the only means to protect your computer investment is to order an 80-GRAFIX board TODAY!



EXCITEMENT & FUN
Open up a new realm of software development with the 80-GRAFIX board.

PROGRAMMA INTERNATIONAL, INC.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579 • 384-1116 • 384-1117

Available exclusively through PROGRAMMA at the cost of $149.95
Please check with us for availability prior to ordering
VISA and MASTERCHARGE accepted
TRS-80 is a registered trademark of the Tandy Corp.

✓21

✓Reader Service—see page 242

```
420 IF X>(A(4)*12)PRINT" - - - - -NEGATIVE CASH FLOW - - - -"
430 PRINT: INPUT"PRESS  /ENTER/  TO CONTINUE";Q:CLS
440 L=A(1)-(A(1)*(A(2)/100))
450 N=L*(A(9)/100)
460 DI=N
470 EB=(M*12)-DI
480 IB=A(1)*A(7)
490 TD=DI+PT+IC+(ME*12)
500 ND=(A(4)*12)-TD
510 TB=ND*(A(6)/100)
520 PRINT"TOTAL PURCHASE PRICE $";A(1)
530 PRINT"TOTAL DOWN PYT (INCL FEES) $";TE
540 PRINT"DOWN PYT IS ";(TE/A(1))*100;" % OF PURCHASE P
    RICE"
550 PRINT
560 PRINT"TOTAL EQUITY = $";TE
570 PRINT
580 PRINT"ITEM","DOLLAR AMT","PERCENTAGE"
590 PRINT"CASH FLOW",(A(4))*12-X,(((A(4)*12)-X)/TE)*100
600 PRINT"EQUITY BUILD UP",EB,(EB/TE)*100
610 PRINT"INFLATION";A(7);"%",A(1)*(A(7)/100),((A(1)*(A
    (7)/100))/TE)*100
620 PRINT"TAX BENEFIT",TB,(TB/TE)*100
630 DA=((A(4)*12)-X)+EB+(A(1)*(A(7)/100))+TB
640 PR=(DA/TE)*100
650 PRINT
660 PRINT"TOTALS",DA,PR
670 PRINT
680 INPUT"PRESS  /ENTER/  TO MODIFY ENTRIES";Q:CLS:GOTO
    170
```

Program Listing 1

SAMPLE PROGRAM

INCOME PROPERTY EVALUATOR I

1.	PURCHASE PRICE $	12000
2.	DOWN PAYMENT %	15
3.	ADDITIONAL CLOSING COSTS $	400
4.	MONTHLY RENTAL INCOME $	240
5.	LAND VALUE (% OF PURCHASE PRICE)	10
6.	TAX BRACKET %	23
7.	PROPERTY INFLATION RATE %	4
8.	YEARS OF 1ST MORTGAGE	30
9.	INTEREST RATE ON 1ST MORTGAGE %	12

IF ALL ENTRIES ARE CORRECT ENTER #10, IF NOT ENTER THE # TO BE CORRECTED? ______

PURCHASE PRICE $ 12000 TOTAL DOWN PYT $ 2200 WHICH IS 18.3333 %

ITEM	ANNUAL AMOUNT
MORTGAGE PYT	1259.02
PROPERTY TAX	158.4
INSURANCE	99
MGT FEE/MAINT.	432
RENTAL INCOME	2880
CASH FLOW	931.577

PRESS /ENTER/ TO CONTINUE? ______
TOTAL PURCHASE PRICE $ 12000
TOTAL DOWN PYT (INCL FEES) $ 2200
DOWN PYT IS 18.3333 % OF PURCHASE PRICE
TOTAL EQUITY = $ 2200

ITEM	DOLLAR AMT	PERCENTAGE
CASH FLOW	931.577	42.3444
EQUITY BUILD UP	35.0233	1.59197
INFLATION 4 %	480	21.8182
TAX BENEFIT	222.318	10.1054
TOTALS	1668.92	75.8599

PRESS /ENTER/ TO MODIFY ENTRIES? ______

program takes a very conservative approach to evaluation.

Line 270 sets the insurance cost. The formula shown is for the type and amount of insurance I carry on my properties. The insurance industry uses many complex formulas to come up with an insurance rate. The best way to find your annual cost is to visit your insurance agent, explain that you are considering several properties and ask for a rate on the type of property you desire. Once you have the dollar amount per year, you can figure the variable to use in the evaluations. The variable equals yearly rate divided by purchase price times one thousand.

All you need do now is decide if you are going to collect the rents or let a property management company collect them for you (this expense is tax deductable). If you are going to let a company collect the rents, call several for commission charges. Locally, the commissions are ten percent of the rent collected. I also set aside five percent of collected rent for repairs. Line 310 sets the management expense at 15 percent of collected rents.

There are only two more pieces of information that you need, the closing costs that you will have to pay (ask a realtor) and your tax bracket. You are now ready to start evaluating property.

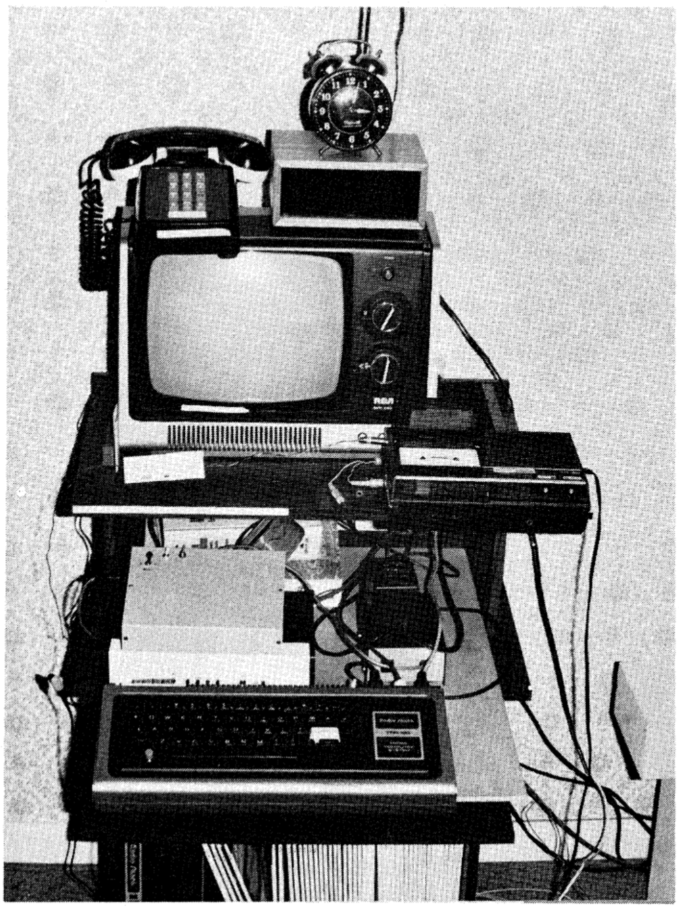
The property shown in the sample program is one that I recently purchased. As you see, I get a 75 percent return on my investment the first year. One of my requirements is a minimum of 60 percent return. The owner wanted $12,000 with 50 percent down, and he would carry a note for 10 years at 14 percent. This would make my investment $6400 with only a 33 percent return. I offered $8000 with 20 percent down, leaving the terms the same, giving me a 100 percent return. The owner was set on his price of $12,000, so after many offers we settled on the terms shown in the sample program. If the owner won't change the price, start working on the terms.

If you are new to real estate investing, one rule to remember is that nothing is final until signed, and once signed it is final.

This program works equally well for buyers or sellers. ■

For those who have more time than money.

Electronic Systems Serial I/O Board



The author's system. The TV is driven by an RF modulator enclosed in the case that contains the Electronic Systems TRS-80 I/O board and power supply. This case is located just behind the keyboard on top of the A 103 modem.

TRS-80 Serial I/O Board
Electronic Systems
San Jose, CA
$79.95 assembled
$59.95 kit

by Jim Cambron
P.O. Box 10005
Kansas City, MO 64111

The spring of '79, electronic message system fever struck Kansas City. It started with the Computer Network of Kansas City's Electronic Message System, which, unfortunately, disappeared after a promising showing. The fever was later renewed with the appearance of Bill Abney's FORUM-80 system. It was soon discovered that there were quite a few electronic message systems up and running around the country—most notably the handiwork of The Peripheral People (ABBS on Apple) and Ward Christensen/Randy Seuss (CBBS on S-100 mainframes).

And here I was with a Level II 16K TRS-80, and not much bread to spend on telecommunications.

As an alternative to an expansion interface and an RS-232 board (which is in the neighborhood of $350), I purchased the TRS-80 Serial I/O board without parts from Electronic Systems. I scrounged in my junk box for parts and ended up spending around $65, including the power supply and case.

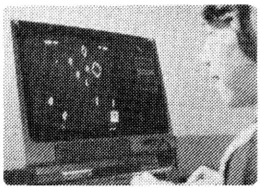
You can buy a complete kit (without power supply or case) or the assembled unit (no power supply or case). The bare board costs $19.95. Cables and connectors to the TRS-80 expansion bus and the modem of your choice are also extra. Even with the extras the cost is only about $150, and the darn thing works!

Before you go running out to buy this serial interface, you should realize there are some sacrifices you will be making for the sake of saving some money. The TRS-80 Serial I/O will not work with any Radio Shack terminal software. The TRS-80 I/O board uses port 37H for input and address 37F8H for output.

By the time you read this, by the way, Electronic Systems will be offering an updated version of the TRS-80 Serial I/O board including at no exta cost a crystal controlled oscillator circuit. This modification will improve the reliability of the unit.

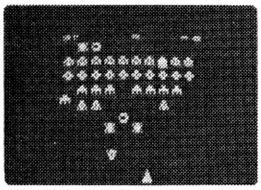
Games from BIG FIVE will turn your computer into a

TRS-80 HOME ARCADE



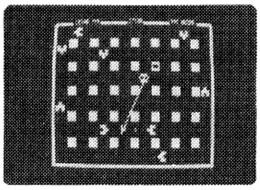
SUPER NOVA©

If you and your TRS-80 have longed for a fast-paced arcade-type game that is truly a challenge, then **SUPER NOVA** is what you've been waiting for. In this two player machine-language game, large asteroids float ominously around the screen. Suddenly your ship appears and you must destroy the asteroids before they destroy you! (But watch out because big asteroids break apart into little ones.) The controls that your ship will respond to are thrust, rotate, hyperspace, and fire. All right! You've done it! You've cleared away all the asteroids! But what is that saucer with the laser doing? Quick! You must destroy him fast because that guy's accurate!



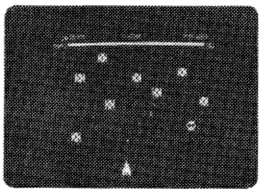
GALAXY INVASION©

The sound of the klaxon is calling you! Cruel and crafty invaders have been spotted in battle formation warping toward Earth at an incredible speed. Suddenly, your ship materializes just below the huge flock of invaders. Quickly and skillfully you shift right and left as you carefully fire your lasers at them. But watch out! A few are breaking out of the convoy and flying straight at you! As the whine of their engines gets louder, you place your finger on the fire button knowing all too well that this shot must connect—or your mission will be permanently over! With sound effects!



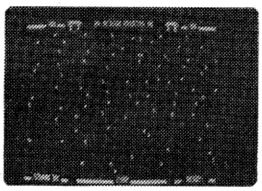
ATTACK FORCE©

Your TRS-80 screen has been transformed into a maze-like playfield for this game. As your ship appears on the bottom of the screen, eight alien ramships appear on the top. All of them are traveling at flank speed directly at you! Quickly and boldly you move toward them and fire missiles to destroy them. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the wrath of the keeper of the mazefield: the menacing "Flagship". You must destroy him fast because, as you will find out, that guy's accurate! With sound effects!



COSMIC FIGHTER©

With thousands of stars whizzing by you, your **SPACE DESTROYER** ship comes out of hyperspace directly under a convoy of aliens. Almost effortlessly, you skillfully destroy every last one. But before you can congratulate yourself, another set appears. These seem to be slightly more intelligent than the first set. Quickly you eliminate all of them, too. But your fuel supply is rapidly diminishing. You must still destroy two more sets before you can dock with your space station. All right! The space station is now on your scanners! Oh no! Intruders have overtaken the station! You must skillfully fire your neutron lasers to eliminate the intruders from the station before your engines run out of fuel and explode! With sound!



METEOR MISSION II©

The second **Big Bang** has occurred and the galaxy is full of stray asteroids and meteors. As you look through your space port you see a belt of asteroids drifting across the screen blocking your path to the safety of the space station above. But be careful because meteor showers, exploding suns and invading aliens may strike your ship and send it hurtling back to ground level. How many times can you and your opponent maneuver through those obstacles before time runs out? With sound effects!

BIG FIVE SOFTWARE

✓357

P.O. Box 9078-185, Van Nuys, California 91409

Prices per game: Level 2, 16K Mod 1/Mod 3—$15.95
Level 2, 16K Mod 1/Disk—$17.95
10% discount for 2 games, 15% for 3 or more.
Please add $1.50 postage/handling, Calif. residents add 6% tax.
All games are written in machine language and supplied on cassette.
Disk versions save high scores to your TRSDOS or NEWDOS diskette.

Cassette versions require 16K memory, disk versions require 32K.
We accept checks, money orders, and MC/Visa orders.
All games ©1980 by Bill Hogue & Jeff Konyu.
TRS-80 & TRSDOS are trademarks of Tandy Corp.
NEWDOS is a trademark of Apparat, Inc.
Dealer inquiries invited.

Give Card Number, Expiration Date and Signature for Master Charge and VISA orders.

Circle R/S number 357 or write us for info on new JOYSTICK versions.

✓Reader Service—see page 242

Assembly

Assembly of the circuit board is straightforward. Although the instructions are brief, the average person with some knowledge of soldering can assemble it in an evening. However, there are some things to watch out for at various points in the assembly. For example, be sure that you use a 74C93 (CMOS device) for U-9; my instructions called for a 7493, but I discovered from a friend who had ordered a kit with parts that a 74C93 has a different pinout.

Also, note that the clock oscillator circuit works better using 74C04 CMOS hex inverters than with the specified 7404 ICs.

There was also some confusion as to which wires went where on the RS-232 connector. Pin 3 of the DB-25 connector is wired to the IN pad on the board, Pin 2 is wired to the OUT pad, and Pin 7 to the GND pad. The best way to set the oscillators is to connect a frequency counter to the output of each circuit and adjust the trim pots as instructed. This method guarantees accurate clocking of the UART.

Once I set my unit up, I never had to go back and readjust anything.

Some careful consultation with Bill Abney resulted in Table 1, which describes a workable switch configuration for communicating with electronic message systems. You may be interested to know that the FORUM-80 systems will display TRS-80 graphics if you program your UART for eight-bit words. To do this, I ran a switch outside to the case that lies parallell to switch two on the UART pro-

Baud rate: switch 1 on = 110 BAUD
3 on = 300 Baud

UART Programming dipswitch setup for terminal communication:

Switch 1 off
(Parity off = even on = odd)

Switch 2 on
(word length 8u-bits = off 7-bits = on)

Switch 3 off
(word length select switch)

Switch 4 on
(Stop bits 1 = on 2-off)

Switch 5 off
(Parity = on No Parity = off)

Table 1. Switch Programming Table

Program Listing

```
                00100 ;        DOWNLOAD UTILITY   ESIO   VERSION 16K LEVEL II
                00110 ;        SUGGESTED NAME =  EDLOAD
                00120 ;        WRITTEN BY JIM CAMBRON (WITH HELP FROM BILL ABNEY)
                00130 ;        FOR USE WITH THE ELECTRONIC SYSTEMS TRS-80 SERIAL
                00140 ;        I/O BOARD.  THIS PROGRAM PERMITS PROGRAM TRANSFER
                00150 ;        (DOWNLOADING) FROM A FORUM-80 SYSTEM TO YOUR
                00160 ;        SYSTEM THROUGH THE 'X' COMMAND ON FORUM 80.
                00170 ;        NOTE:   THIS TERMINAL PROGRAM PATCHES THE SERIAL
                00180 ;        TRS-80 SERIAL I/O INTERFACE INTO THE KEYBOARD
                00190 ;        SCAN ROUTINE WHILE OPERATING UNDER THE BASIC
                00200 ;        INTERPETER.  WHEN DOWNLOADING A PROGRAM, PLEASE
                00210 ;        REMEMBER TO CONFIGURE FOR AT LEAST 25 NULLS SO
                00220 ;        THE BASIC INTERPETER CAN KEEP UP WITH THE
                00230 ;        PROGRAM TRANSFER.  TO CONFIGURE FOR 25 (OR MORE)
                00240 ;        NULLS ON THE FORUM-80, ENTER A 'C' COMMAND.
                00250 ;        ONCE YOU HAVE DONE THIS, ENTER AN 'X' COMMAND
                00260 ;        TO ENTER THE PROGRAM TRANSFER MODE.
                00270 ;
                00280 ;        THIS TERMINAL PROGRAM WILL ONLY WORK WITH
                00290 ;        LEVEL II BASIC ONLY!!!
                00300 ;
                00310 ;        THIS PROGRAM CAN BE RELOCATED IN SOURCE BY
                00320 ;        CHANGING THE VALUE OF THE FOUR-DIGIT HEX VALUE
                00330 ;        IN LINE 410 TO MEET YOUR NEEDS IN RE-ASSEMBLING.
                00340 ;        YOU MUST PROTECT THIS PROGRAM BY SPECIFYING A
                00350 ;        MEMORY SIZE VALUE LESS THAN THE STARTING ADDRESS
                00360 ;        OF THIS PROGRAM OR HORRIBLE AND FRUSTRATING
                00370 ;        THINGS WILL HAPPEN!  ALWAYS RELOCATE TO THE TOP
                00380 ;        OF YOUR MEMORY SO YOU WILL HAVE PLENTY OF MEMORY
                00390 ;        FOR THE BASIC PROGRAM YOU DOWNLOAD.
                00400 ;
7F6C            00410          ORG     7FFFH-93H        ; CHANGE TO MEET MEM SIZE
                00420 ;
                00430 ;                                   SYMBOL TABLE
0033            00440 CRT      EQU     33H       ; CRT = DISPLAY ROUTINE
03E3            00450 KEYIN    EQU     03E3H     ; KEYIN = KEYBOARD SCAN ROUTINE
                00460                            ; (ADDRESS CALLED BY BASIC)
002B            00470 KBD      EQU     2BH       ; KBD = KEYBOARD SCAN ROUTINE
                00480                            ; (PART OF KEYIN ROUTINE)
4016            00490 KDCB     EQU     4016H     ; ADDRESS OF KEYBOARD DEVICE
                00500                            ; CONTROL BLOCK (DCB)
                00510 ;
000A            00520 STOR     DEFS    0AH              ; REGISTER STORAGE
                00530 ;
                00540 ;        START OF PROGRAM
                00550 ;
7F76 01807F     00560 DLOAD    LD      BC,DPATCH        ; CHANGE DCB
7F79 ED431640   00570          LD      (KDCB),BC        ;
7F7D C3191A     00580          JP      1A19H            ; JUMP TO BASIC
                00590 ;
                00600 ;        DCB PATCH ROUTINE.
                00610 ;
7F80 01F837     00620 DPATCH   LD      BC,37F8H         ; INITIALIZE..
7F83 0A         00630          LD      A,(BC)           ; GET STATUS
7F84 E608       00640          AND     08H              ; OF MODEM
7F86 2804       00650          JR      Z,KEY            ; NOTHING, CHECK KEYBOARD


7F88 ED78       00660          IN      A,(C)            ; INPUT A BYTE
7F8A 1803       00670          JR      CONT             ; AND JUMP TO CONT.
7F8C CDE303     00680 KEY      CALL    KEYIN            ; CHECK KEYBOARD
7F8F B7         00690 CONT     OR      A                ; IS IT ZERO?
7F90 2808       00700          JR      Z,CONT1          ; RETURN TO INTERPETER
7F92 FE0A       00710          CP      0AH              ; IS IT A LINEFEED?
7F94 28F6       00720          JR      Z,KEY            ; YES, CHECK KEYBOARD
7F96 FE14       00730          CP      20D              ; IS IT EOF CHARACTER?
7F98 2801       00740          JR      Z,EXIT           ; YES, EXIT PATCH
7F9A C9         00750 CONT1    RET                      ;RETURN TO INTERPETER
7F9B 01E303     00760 EXIT     LD      BC,KEYIN         ; RESTORE DCB
7F9E ED431640   00770          LD      (KDCB),BC        ; TO FORMER STATE
7FA2 ED536C7F   00780          LD      (STOR),DE        ; SAVE DE
7FA6 226E7F     00790          LD      (STOR+2),HL      ; SAVE HL
7FA9 DD22707F   00800          LD      (STOR+4),IX      ; SAVE IX
7FAD FD22727F   00810          LD      (STOR+6),IY      ; SAVE IY
                00820 ;
                00830 ;        TERMINAL ROUTINE
                00840 ;
7FB1 CDC901     00850 TERM     CALL    1C9H             ; CLEAR SCREEN
7FB4 3E0E       00860          LD      A,14D            ; CURSOR ON
7FB6 CD3300     00870          CALL    CRT              ;
7FB9 01F837     00880          LD      BC,37F8H         ; INITIALIZE ESIO
7FBC ED78       00890          IN      A,(C)            ; RESET UART
                00900 ;
7FBE CDDF7F     00910 LOOP     CALL    INPUT            ; CHECK INPUT
```

Program continues

Reader Service—see page 242

```
7FC1 FE12               CP      18D             ; OPEN FILE CHAR?
7FC3 2809               JR      Z,TDLOAD        ; YES, INSTALL PATCH
7FC5 CDED7F             CALL    OUTPUT          ; CHECK OUTPUT
7FC8 FE12               CP      18D             ; OPEN FILE FROM KBD?
7FCA 2802               JR      Z,TDLOAD        ; YES, INSTALL PATCH
7FCC 18F0               JR      LOOP            ; OTHERWISE LOOP BACK
7FCE ED5B6C7F   TDLOAD  LD      DE,(STOR)       ; RESTORE DE
7FD2 2A6E7F             LD      HL,(STOR+2)     ; RESTORE HL
7FD5 DD2A707F           LD      IX,(STOR+4)     ; RESTORE IX
7FD9 FD2A727F           LD      IY,(STOR+6)     ; RESTORE IY
7FDD 1897               JR      DLOAD           ; JUMP TO PATCH
                ;
                ;
7FDF 0A         INPUT   LD      A,(BC)          ; GET STATUS
7FE0 E608               AND     08H             ; CHECK IT
7FE2 C8                 RET     Z               ; NOT READY, RET
7FE3 ED78               IN      A,(C)           ; INPUT BYTE
7FE5 FE12               CP      18D             ; OPEN FILE CHAR?
7FE7 C8                 RET     Z               ; YES, RET.
7FE8 CD3300             CALL    CRT             ; OTHERWISE, DISPLAY
7FEB 18F2               JR      INPUT           ; LOOK FOR ANOTHER.
                ;
                ;
7FED CD2B00     OUTPUT  CALL    KBD             ; CHECK KEYBOARD
7FF0 B7                 OR      A               ; NOTHING?
7FF1 C8                 RET     Z               ; YES, RETURN
7FF2 FE12               CP      18D             ; OPEN FILE CHARACTER?
7FF4 C8                 RET     Z               ; YES, RET.
7FF5 F5                 PUSH    AF              ; SAVE AF
7FF6 0A         SEND    LD      A,(BC)          ; GET STATUS
7FF7 E624               AND     24H             ; AND CHECK IT
7FF9 28FB               JR      Z,SEND          ; NOTHING, CHECK AGAIN
7FFB F1                 POP     AF              ; RESTORE AF
7FFC 02                 LD      (BC),A          ; OUTPUT BYTE
7FFD C9                 RET                     ; RETURN
                ;
7F76                    END     DLOAD
00000 TOTAL ERRORS
```

gramming dipswitch. Now, I switch from seven to eight-bit mode easily.

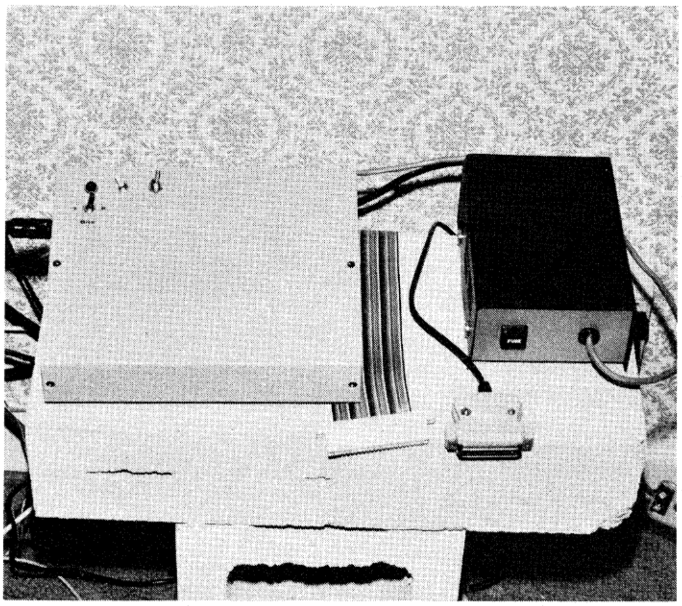
The TRS-80 I/O board comes with a machine language line-printer routine, terminal program and a routine to send BASIC programs to another computer. The latter program didn't work, but I figure two out of three isn't bad. I assembled the terminal program using Radio Shack's editor assembler and made my first call to another computer within minutes without a hitch!

Several of the various electronic message systems now offer a way to transfer public domain BASIC programs to your system so that you can save them and run them at your leisure. There are several excellent terminal software packages, including Lance Miklus's ST80 III. This is available to transfer programs.

As far as I know, ST80 III is not available for use on the Electronic Systems TRS-80 I/O board.

The program listing, however, is a source listing of a terminal program that will allow you to transfer BASIC programs with the Electronic Systems TRS-80 I/O board. The terminal program contains two parts, a standard terminal program and a routine that is patched into the keyboard scan routine through the keyboard device control block (DCB).

To use the program, you must power up and specify a memory size of 32600. Enter the stystem command and load the program. Execute by entering slash (/), and then enter a CLEAR command. This is important. Otherwise, the first attempt to enter any other BASIC statement will result in an MO ERROR. Next press the shift, down arrow and T keys to jump to the terminal mode. The screen should clear and a cursor will appear at the top right of the screen. Press the shift, down arrow, and R keys to return to BASIC to SAVE or RUN a program.



Electronic Systems TRS-80 I/O board enclosed in a sloped cabinet from Radio Shack. DB-25 connects to modem (or serial printer); ribbon cable connector plugs in to the keyboard. In the upper, left hand corner are (left to right) the LED power on indicator, on-off switch, address change switch; and below the LED indicator is 7-bit/8-bit switch.

When attempting to transfer a BASIC program with this terminal program, request at least 25 nulls be sent by the system called at the beginning of each line. The nulls allow the BASIC interpreter in read-only-memory (ROM) time to process each line transmitted. Follow the system's instructions on how to change the number of nulls being sent. It's usually done by entering the C for change command or N for nulls.

When you have successfully told the system to add the nulls, enter the program transfer or download utility by typing the appropriate command character (L on the FORUM-80). A menu of programs will be displayed. Choose the number for the program you want and enter it. The system will control your computer and transfer the specified program automatically. You can at any time override the FORUM-80 systems by pressing the shift, down arrow, and T keys to return to the terminal mode, followed by the character S which stops the transfer function. When you have retrieved the program, you can return to BASIC and RUN or SAVE the program.

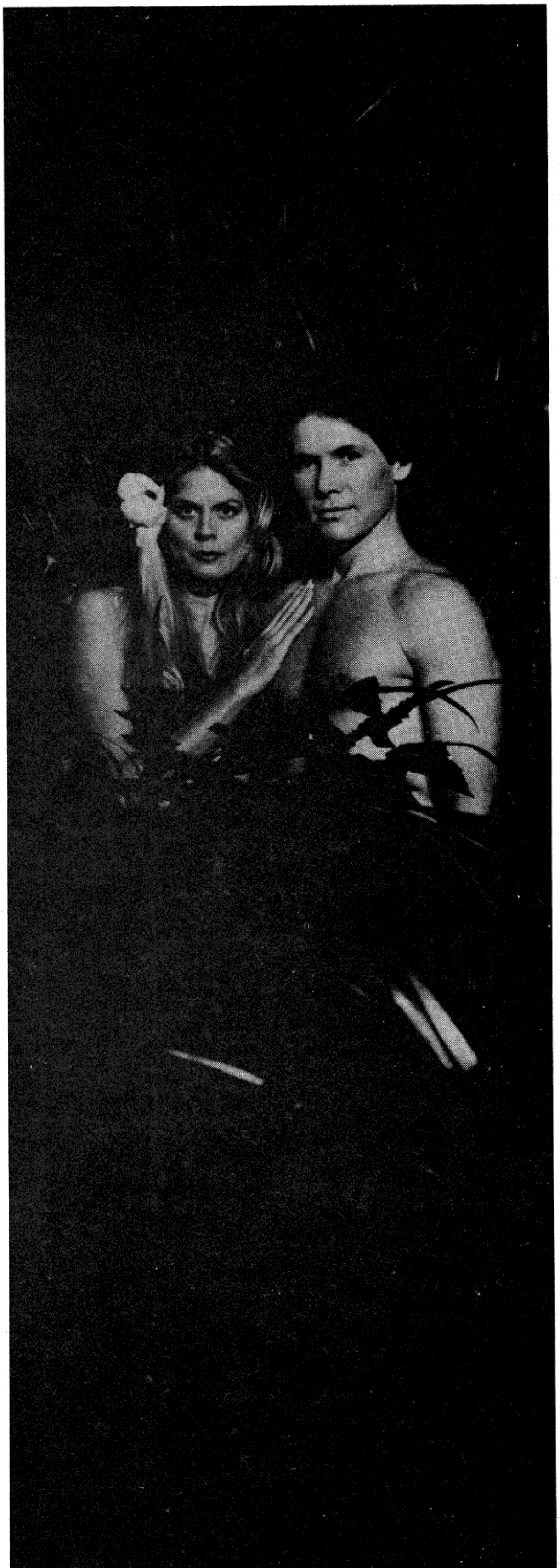
The Electronic Systems TRS-80 Serial I/O Board is great for those who have time to tinker and are also short on cash. I have not yet touched on the product's usefulness as a serial printer interfacing device, which may be *your* reason for purchasing it. But, I hope you find it as useful and educational as I have.■

in the Beginning

there was the word.

And the word was BASIC

For years computer programmers relied on a series of complex coded language to create programs for a host of computers. While these languages made the computer operate at lightning speeds, they did limit the ability of the average person to initiate computer programs. Then came the microcomputers and with them a return to BASICS. Now programs could be written using very simple language and the individual could communicate with the computer in plain English.



INCREASE PROGRAMMERS SPEED

With the introduction of the TRS-80 family of computers, a new world opened to the average programmer. SNAPP, INC. saw the potential for a more advanced form of basic programming that allowed the programmer to do more and at the same time increase productivity. Thus was created "SNAPP II"; a family of enhancements to the Model II BASIC interpreter. While part of the package originated with the best of APPARAT, INC.'S thoughts in implementing NEWDOS BASIC, SNAPP, INC. integrated some exciting new enhancements that won praise from Model II owners and programmers alike.

SNAPP, INC. created a system that demonstrates the company as the country's foremost expert on software for TRS-80's.

SNAPP II is written entirely in machine language for super fast execution. Its extensions are fully integrated into Model II BASIC and require NO user memory and NO user disk space.

The SNAPP II extended BASIC package is composed of six modules, each of which may be purchased separately.

XBASIC. This module has six single keystroke commands to list the first, last, previous, next or current program line or to edit the current line. Includes a quick way to recover BASIC program following an accidental re-boot. There are ten single character abbreviations for frequently used commands; AUTO, CLS, DELETE, EDIT, KILL, MERGE, NEW, LIST and SYSTEM.

XREF: A module that contains a powerful cross-reference facility with output to display and/or print. You can easily trace a variable through the code and also determine if a variable is in use.

XDUMP: The programmer is permitted, through this module, to display and/or print the value of any or all program variables. It identifies the variable type for all variables with each element of any array listed separately.

XRENUM: An enhanced program line renumbering facility which allows specification of an upper limit of the block of lines to be renumbered. Supports relocation of renumbered blocks of code, and duplication of blocks of code.

XFIND: This module uses a cross-reference facility for quick and easy location of specified strings or keywords within the program text. Also includes global replacement of keywords.

XCOMPRESS: Compress your BASIC programs to an absolute minimum. Removes extraneous information; merge lines; even deletes statements which could not be executed. Typically saves 30-40% space even for programs without REM statements! Also results in 7-10% improvement in execution speed.

While all of the above modules are the best on the market, SNAPP, INC. is continually striving to upgrade programmers capabilities and productivity. If you are a programmer using a Model II without the SNAPP II family of extended BASIC enhancements, you are operating below your own capacity. Make your TRS 80 system truly cost-efficient.

Get more from your Model II
with SNAPP II.
Individual Modules $40.00
Entire package only $200.00
Now available for the Model III
Individual modules $25.00
Entire package only $125.00

Here's how to buy it. Credit card customers add 2%. C.O.D. customers add $2.00. Ohio residents add 4½% sales tax. Shipments normally made the same day we receive your order. Credit granted to governmental agencies and educational institutions and D & B rated business firms. Please include purchase order number when ordering.

Here's our guarantee. If your diskette arrives damaged, we will replace it without charge. If you accidentally damage it, we will replace it for a $10.00 handling charge. For one year we will provide you with any enhancements or updates for a $10.00 handling charge. If the remote chance that errors are discovered in the program, they will be corrected for free for a period of one year. If we can't correct it, return program for full refund.

TRS 80 is a trademark of the Radio Shack division of Tandy Corporation.

NEWDOS AND NEWDOS/80 are trademarks of APPARAT, INC.

SNAPP, INC.

8160 Corporate Park Dr. ✓232
Cincinnati, Ohio 45242
Call Toll Free
1-800-543-4628
Ohio residents call collect
(513) 891-4496

SAVE
on add-ons for
TRS-80®
Software & Hardware

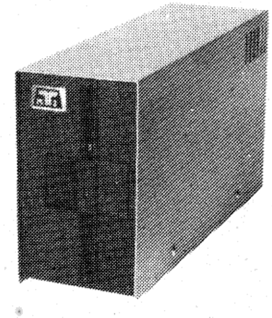
NEWDOS80
A new enhanced NEWDOS for the TRS-80®

The most powerful Disk Operating System for the TRS-80, designed for the sohpisticated user and professional programmer who demands the ultimate.

NEWDOS/80 is the planned upgrade from NEWDOS 2.1. Some of the features are:

- New BASIC commands for files with variable record lengths up to 4095.
- Mix or match drives. Use 35, 40 or 80 track 5" disk drives or 8" disk drives, or combo.
- Security boot-up for BASIC or machine code application programs.
- New editing commands.
- Enhanced RENUMber that allows relocation.
- Command chaining.
- Device handling for routing to display and printer simultaneously.
- DFG function; striking of D, F and G keys allows user to enter a mini-DOS without disturbing program.
- Compatible with NEWDOS & TRSDOS.
- Machine language Superzap/80 2.1 utilities ad enhanced debug and copy.

$149



FACTORY CLEARANCE

Demo single or dual head MPI disk drive, complete with Power Supply and Chassis. Full warranty.

TF5D 40 TK. **$279**

TDH-1D Dual 35 TK **$419**

LIMITED QUANTITIES

MONTHLY SPECIAL
Shugart SA801 Bare Drive **$469**

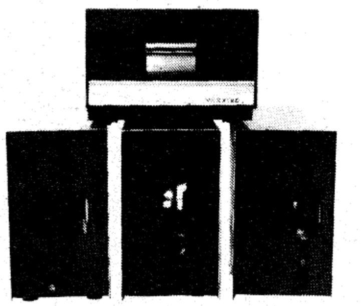
NEW FOR TRS-80®

TF-8 80 TRACK DISK DRIVE by MPI
Double your capacity. Single head mini floppy. More than 200K bytes of storage. Complete with power supply and chassis.
$639

TF-9 DUAL 80 TRACK DISK DRIVE by MPI
Quadruple your capacity. 400K bytes of storage (like having 4-40 track drives in one unit). Complete with power supply and chassis.
$789

Disk Drive Sale!
Complete with power supply and chassis.

TF-3 Shugart SA400	$329
Shugart SA410 40 tk.	$339
Pertec FD200, 40 track	$379
TF-5 MPI B51, 40 track	$359
TF-7 Micropolis 77 track	$579
TDH-1 Dual sided drive 35 track	$469
TF-3 Drive sys. 2 Shugart 40 tk.	$658
NEWDOS+ 40 tk. ..... $110 30 tk.	$99
AJA Business Pkg.	$359
The Source	$100
Basic Compiler	$195



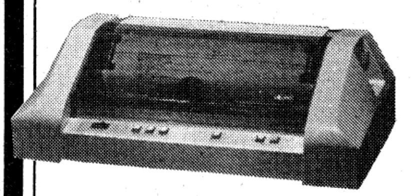
® Registered trademark of Radio Shack

Disk Expansion System

• 2 Shugart SA400 TF-3	$678
• 1 Two-Drive Cable	$26
• 1 Expansion Interface 32K	$459
• 1 35-track DOS+	$99
TOTAL LIST PRICE	$1262
SPECIAL PRICE ONLY	**$1,149**

MOD II 8" Disk System

• 1 Drive System	$949
• 2 Drive Expansion System	$1,445



Printers

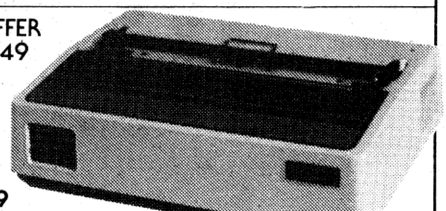
Centronic 700	$1,069
Base 2	$649
Centronics 737	$895
Centronics 702-9	$1,995
Malibu	$2,495
Paper Tiger	$949
Daisy Wheel	$1,779

Anadex 9500 .... $1,549

INTRODUCTORY OFFER
SAVE $300 LIST $949

Okidata Microline 80
$639

Model 82 ...... $899
Model 83 .... $1,195



★ BARE DRIVES FOR ANY MICROCOMPUTER

Pertec FD200	$282	FD250	$359
Shugart SA400 35 track	$269	SA410 40 track	$279
MPI B51	$279	B52	$349
MPI B91	$399	B92	$525

Memory Kit (16K)	$49.00
AC Isolator (6 socket)	$49.95
Disk Head Cleaner	$19.95
Diskettes Verbatim (10)	$30.00

MICROCOMPUTER TECHNOLOGY INCORPORATED

✓28

3304 W. MacArthur
Santa Ana, CA 92704
(714) 979-9923

Order Desk Only 800-554-7222
Telex #678401TAB IRIN

VISA · MasterCard

ALL PRICES CASH DISCOUNTED
FREIGHT FOB FACTORY
ASK FOR FREE CATALOG

More on interfacing BASIC and Machine Code.

Be a USR User

Allan S. Joffe W3KBM
1005 Twining Rd.
Dresher, PA 19025

My first contact with USR(0) left me a bit puzzled. The USR(0) command allows you to call a machine language program from any BASIC listing. But I didn't find the explanation for it in the Level II manual to be a model of clarity.

Besides, I thought, if I'm going to insert a machine language program, I'm surely going to treat myself to a copy of T-BUG or a similar program. Why mess around with the POKE routine? T-BUG is both faster and easier.

On the Doorstep

Since using USR(0) puts you on the doorstep of machine or assembly language, you must acquaint or reacquaint yourself with hex.

The first order of business is to determine the decimal addresses 16526 and 16527 in hex: 408E and 408F. These locations hold the address where you will insert your new machine language program.

Getting In

With the selection of an origin or starting address of 4A00 we can determine what we should now load into memory locations 408E and 408F. These locations will contain 00 4A in that order via T-BUG or the monitor program of your choice. If this seems backwards, don't try to argue with the logic of the computer! Backwards or not, the machine will digest the command properly.

Assuming that you have loaded in T-BUG and changed the two locations as shown, perform a J 1A19 with T-BUG. This performance will get you READY, so that you can insert a short BASIC program, utilizing USR(0).

Prior to going back to BASIC, however, let's load in the following, short, assembly language program. Let's start with memory location 4A00.

```
16 BF 21 00 3C 01 00 04 72 23
0B 78 B1 20 F9 C9
```

"Painting" the Screen

The basis for this program may be found on page 100 of William Barden's book, *The Z-80 Microcomputer Handbook*. It paints the screen, or turns on the entire screen, when it is called with the USR(0).

The second byte, BF, is hex for the graphics block that you are familiar with as 191 in decimal. You've probably run a routine at some time or other that painted the entire screen. And you most likely used the SET or PRINT commands with strings of characters. If you've done this, you know how slow the first method is and how fast the second is just by comparing the two.

Let's return to BASIC now and input the following program:

```
10 For T = 1 to 300:NEXT T
20 J = USR(0)
30 FOR T = 1 TO 300:NEXT T
40 CLS
50 GOTO 10
```

When you RUN this, your screen flashes on during the timing loop and then it goes dark during the next one. This repeats itself until you BREAK the program.

It is an excellent demonstration of just how fast machine language is next to BASIC when presenting graphics on the screen.

It may be of passing interest to note that there is no magic in the 0 in USR(0). You can see this by making the instruction read USR(7), in fact, any alphanumeric and the USR command might do nicely.

Now try this BASIC program for a bit of a change. Use the machine language program still at 4A00.

```
10 FOR X = 1 TO 50
20 PRINT X;
30 FOR T = 1 TO 100:NEXT T
40 W = USR(0)
50 NEXT X
```

Note the effect of the numbers marching across the screen. It's now a totally white screen and the number is surrounded by a black box.

Next, let's make a small change. We return to T-BUG and change the second byte (which is currently BF) and make it a 20. This is the hex code for a blank.

If we return again to BASIC and RUN (remember, we still have our last BASIC program resident in memory), we'll see the numbers flash on the screen. But this time the entire screen is black except for the number currently printed out. If you think about that, the program now resident at 4A00 is effectively a CLS command and it may have a use as such.

That use is this: if you want to program a display using the large print mode (32 char/line), include a regular CLS anywhere. When it is executed, it will then drop you back into the small print mode. By using the machine language routine the CLS function is available that then lets you get to the 32 char/line if that is your desire.

As you have already noticed, the USR(0) program has been configured as a subroutine. It functions in a manner similar to the subroutine that you are used to in BASIC.

ADD CLS to T-BUG

This CLS routine can also be added to T-BUG. When using T-BUG, and the routine is executed often, it is necessary to return to T-BUG resulting in garbage video on the screen.

If we insert the following routine, starting at memory location 484E of T-BUG, and punch a copy of the new T-BUG, we now have a T-BUG plus CLS feature added by executing a J484E.

Here is the routine which is inserted starting with location 484E.

```
00  16  20  21  00  3C  01  00  04  72
23  0B  78  B1  20  F9  08  C3  80  43
```

Punch your new copy by entering : P 4380 4975 43A0 T-BUG. You now have your new T-BUG plus CLS.

Now we approach the natural question. What about using more than one machine language routine in the BASIC listing? The answer is certainly, but you have to be careful. Resort to the POKE technique as described in the Level II manual.

Consider, first, this BASIC listing:

```
5 CLS
10 POKE 16526,0
20 POKE 16527,74
30 X=USR(0)
40 FOR J= 1 to 500:NEXT J
50 POKE 16526,0
60 POKE 16527,96
70 X=USR(0)
80 FOR J= 1 TO 500:NEXT J
90 GOTO 10
```

Lines 10 and 20 set up a memory location 4A00 as the origin of one machine language program called by USR(0).

Lines 50 and 60 set up memory location 6000 as the origin of the second machine language routine.

Please note that 4A00 and 6000 are hex values and enter the two following routines using T-BUG.

Starting with memory location 4A00, insert the following:

```
16  2A  21  00  3C  01  00  04  72
23  0B  78  B1  20  F9  C9
```

This routine fills the screen with asterisks. Starting at hex memory location 6000, load in the identical routine with one exception. Change the second byte from 2A to BF. This lights up the screen when the BASIC program calls it via USR(0). Now if you return to BASIC and run the program, the screen alternately lights up and fills with asterrisks. This demonstrates that you have two USR(0) routines running.

If you want to see something a bit more dynamic, make the following changes to the BASIC listing:

```
7 G=G+3
8 POKE 18945,G
90 GOTO 7
```

Now run the program. It will step through the alphanumerics, graphics characters and other signs from the Level II ROM. Do a bit of thinking on how line eight operates and you are on your way to making USR(0) part of your vocabulary.

Consider that USR(0) is sort of a halfway house on the road to using machine language. It can spice up your BASIC programs with speed and versatility, and bring you one step closer to being the master of your machine and not the other way around. ■

✓Reader Service—see page 242

In your Z-80, random number sequences are rather predictable. Here's why.

Random Tricks

Gene Perkins
5224 Winifred Drive
Fort Worth, TX 76133

There is something going on behind your back in your TRS-80! It is a process which directly connects dynamic memory, spinning plates and random numbers. You have no control over dynamic memory and spinning plates, but knowing how the process works will give you greater control over the random numbers generated by your computer.

Spinning Plates And Dynamic Memory

I had better explain what I mean by spinning plates. Picture the juggling act where the juggler has a row of thin sticks or rods on each of which he balances a spinning dinner plate. Eventually the first plate slows down and starts to wobble. Just when it seems certain that it will fall, the juggler steps back to the plate, gives it a few quick slaps to increase its speed, then continues setting the rest of the plates spinning.

Now take a closer look at dynamic memory. When dynamic RAM is compared with static RAM, we find that dynamic RAM is cheaper, uses less power, requires fewer connecting components, and may be faster. But static RAM has one advantage: when data is stored in static RAM, it will stay there as long as power is supplied to the circuit.

In a dynamic RAM chip each bit is stored as an electrical charge on a small capacitor. Even the best of capacitors has some leakage, and over time, the data stored in a dynamic RAM chip will fade away. See Fig. 1.

This loss will occur even if power remains on the circuit. Furthermore, this loss may occur in a matter of seconds or less. Clearly, if dynamic RAMs are to be useful, a way has to be found to maintain the data until it is needed. That's where the Z-80 refresh system comes in.

While the Z-80 CPU is running, it is taking time out to send signals to the memory chips which raise the charge on the bits that are 'on'. The bits which are 'off' receive no additional charge, so they remain off.

If you are running a BASIC program, you may easily believe that the BASIC interpreter program is executing some special machine language instructions which cause the Z-80 CPU to continually refresh the data in memory. But this is not so; if you have ever written an assembly language program, you know that you did not include any memory refresh instructions. They don't exist. The job of refreshing memory is handled automatically by the Z-80 CPU chip.

When the CPU fetches the next instruction from memory, it has to decode it to see what must be done. While the decoding is being done by one part of the CPU, no data is sent over the address lines. Another part of the CPU sends refresh signals out over the lines which give a boost to some of the memory cells.

This behind-the-scenes act takes place during every machine code instruction. The CPU must remember where it left off on the last cycle, so that each memory cell will get its booster shot before the data fades away. The Z-80 CPU contains a seven-bit register, called the refresh register, which is incremented on each cycle to keep track of which part of memory is refreshed next.

A Great Act

You now have the connection between dynamic memory and the spinning plates. While the juggler tosses a ball into the air he may reach out and give a wobbly plate a refresh in its kinetic charge while he is waiting for the ball to come down. While the Z-80 is waiting for an instruction to be decoded, it reaches out through the address lines and gives the memory capacitors an additional electrical charge. It's tricky and it's fast but it's a great act.

Random Numbers

Each random number generated by a computer is based on the previous random number. The previous number is transformed in a specific way to produce the next number. This transformation is fixed, repeatable, and predictable. That's why we call them pseudo random numbers: they're not really random at all.

Given the same starting number, or 'seed', your TRS-80 always generates the same sequence of pseudo random numbers.

The TRS-80 Level II stores the seed and each subsequent ran-

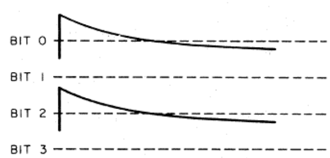
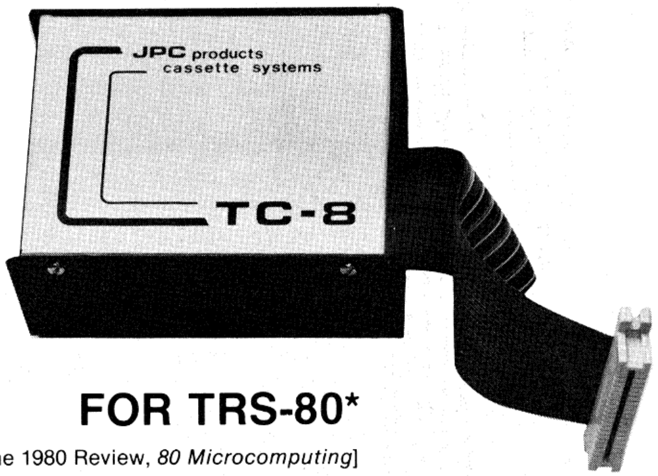


Figure 1.

Poor Man's Floppy

HIGH SPEED CASSETTE SYSTEM



FOR TRS-80*

Now the widely acclaimed JPC Cassette System is available for your TRS-80* computer. The price is only $90.00

[Reprint of June 1980 Review, *80 Microcomputing*]

TC-8 Cassette System
JPC Products
Albuquerque, NM
Kit: $90
Assembled: $120

by Carl A. Kollar

I guess I don't have to tell any TRS-80 owners how frustrating the cassette system that comes with the computer can be. Even with the factory mod that's available, the annoyance of loading and checking programs becomes just barely tolerable.

If you're like me, after you've just plunked down a chunk of money for a Level II 16K machine, "you ain't got nuttin left" for even one disk drive at 500 bucks apiece. So you suffer.

A reasonable alternative is the Exatron Stringy Floppy (ESF). This will cost you about 250 bucks and totally eliminates your loading and saving problems, automatically and fast. I've had one of these for about six months and love it!

But, if the price is still too steep, have I got a device for you!

The Device

The February 1980 issue of *Microcomputing* had an ad that intrigued the hell out of me. It was a high-speed cassette system by JPC Products acclaimed as a "poor man's floppy." It made all sorts of seemingly ridiculous claims such as "loads five times faster," "stores 50,000 bytes on a 10-minute cassette," "less than one bad load in a million bytes with the volume control anywhere between one and eight."

All this for a measly [90] bucks? How could this be? A call to Albuquerque answered a few questions: Yes, it had its own power supply, and, it stored programs five times faster because it utilized higher density data. The computer outputs the information at a higher rate out of the rear keyboard connector.

The ad had even claimed anyone could build it even if you have never soldered before. JPC would make it work, if you couldn't—for free. I was sold. I placed my order, and it arrived about two months later (parts shortage).

I work in electronics, so I found the unit exceptionally easy to build. It took about an hour. The manual is superb. (That's better than great.) It was clear, concise and exact with no ambiguities. Important parts placements are stressed (polarity markings on electrolytics, bands on diodes, etc.).

JPC was right! With these instructions, you couldn't go wrong. The board quality is excellent. It is double-sided and parts locations are clearly marked on the component side of the board. There are no jumper wires to install. JPC utilizes PC traces and plated-through holes for connections to traces on the other side of the board.

Also, there are absolutely no adjustments or settings to bother with.

The documentation is a sheaf of 8½ × 11 papers stapled together. It is written in the nicest format I've seen in a while. Each command and/or subjects is covered on its own sheet in large type. All explanations are in easy to read English—not computerese.

Commands and Features

SAVE"filename": Saves your BASIC program on cassette.
LOAD: Reads the next BASIC program from the cassette.
LOAD"filename": Searches for and loads the specified file from cassette.
LOAD? and LOAD?"filename": Reads file from cassette, and compares contents to memory.
LOADN: Prints a list of all the programs on a cassette, until interrupted by the "break" key.
LOADN"filename": Same as above except the tape will stop at the end of the program named.
KILL: Removes the file manager program from memory so that the extra memory can be used by large programs.
RSET: Allows the operator to rewind and position the tape on tape recorders that have these functions tied to the motor control jack.
RUN"filename": TC-8 searches for a specified program and runs it immediately.
PUT"filename": Same as SAVE "filename", except it is for use with system tapes.
GET: Same as LOAD, except it is for use with system tapes.
GET"filename": Same as LOAD "filename", except it is for use with system tapes.
GET? and GET?"filename": Same as LOAD? and LOAD?"filename", except it is for use with system tapes.
GETN and GETN"filename": Same as LOADN and LOADN"filename", except it is for use with system tapes.
OPEN: Required before cassette input or output of a data file can be attempted.
CLOSE: Required to end a cassette data file.
PRINT#: Allows numerical or string data to be output to a cassette file.
INPUT#: Allows numerical or string data to be input from a cassette file.

I haven't counted them, so I don't know about the "one load in a million bytes" claim, but my son, Anthony (age 11), loaded about 30 of his programs from his Radio Shack format tape to a new TC-8 format tape. He's run them all and found no bad loads.

Unlike the standard tape system, you can position your tape anywhere before the program you want and not have to look for a blank spot between programs. The TC-8 patiently waits for the program you want and then starts loading without getting confused by the portion of the previous program you just fed it.

Try that on your regular cassette system; you'll wear out the reset button. ■

ORDER NOW

To order your TC-8 kit, send your check or money order for $90.00 plus $3.50 postage and handling to JPC PRODUCTS CO., 12021 Paisano Ct., Albuquerque, NM 87112 (New Mexico residents add 4% sales tax). Credit card orders accepted by phone or mail. Personal checks will delay shipment. We will otherwise immediately ship you the TC-8 kit, the cabinet, the ribbon cable, the power adapter, an instruction manual, and a cassette containing the software.

✓190

JPC PRODUCTS CO.
Phone (505) 294-4623
12021 Paisano Ct.
Albuquerque, N.M. 87112

✓Reader Service—see page 242

dom number in memory locations 16554, 16555, and 16556 (in hexadecimal, that is 40AA, 40AB, and 40AC). If you want to compare two processes or programs with the same sequence of numbers, POKE the same seed into those locations at the beginning of the programs. This program will always produce the same numbers:

```
POKE 16554,5:POKE 16555,10:
  POKE 16556,15
FOR I=1 TO 8
PRINT RND(100)
NEXT I
```

No matter when you run it or how many times, you will always get 80, 78, 91, 88, 70, 91, 25, and 30.

Locations 16528, 16529, and 16530 (4090 through4092 in hex) are also used to compute the random numbers. The values stored there are normally 64, 230, and 77. If you don't get the same eight random numbers as before, use the following statement at the beginning of the program: POKE 16528,64:POKE 16529,230:POKE 16530,77.

However, if you are going to play backgammon or poker with a friend on your TRS-80, it may be more fun to get an unforeseen series of events. (It depends on what your motives are.) If we leave it to the electronic circuitry to plant the initial seed in locations 16554 through 16556, numbers may always be the same when you turn on your computer. Most memory cells will initialize to the same value each time power is turned on. You can observe this by noting the characters on your screen each time you turn on the TRS-80, if you have a disk drive attached.

Where do we get a truly random seed? That's where the Z-80 refresh register comes in. There is an assembly language instruction which allows us to access the current value of the refresh register: LD A,R. This loads the value of the refresh register into the accumulator registor A. From there, we can store it anywhere in memory.

When you turn on your TRS-80, the random number seed locations are initialized to some values which will depend on the idiosyncracies of the RAM chips in your system. In a game program, include the RANDOM statement near the beginning. This causes the ROM interpreter to store the current contents of the refresh register into location 16555 (40AB hex) by executing the following ROM routine at address 01D3:

```
LD A,R
LD (40ABH),A
RET
```

The value in the refresh register isn't always the same after power up because the Z-80 starts cycling that register as soon as it is turned on. Since the time from power on until running the program will always be different, the register will always cycle differing amounts, providing a truly unpredictable seed. Even if the register always starts at the same value and your program is automatically loaded and executed from disk, the elapsed time will vary considerably. Differences in temperature and disk or tape position will also ensure that the seed is different for each run.

Don't try to use the refresh register repeatedly by including the RANDOM statement inside a loop. The program in Program Listing 1 illustates what can happen. If you type in the program and run it, you will see that the numbers are not random; they occur most frequently at equal intervals, equally spaced between 0 and 127 because the same number of CPU operations are carried out each time through the FOR-NEXT loop.

Program Listing 2 gives a more random distribution of numbers on the plot. An interesting distribution is obtained by adding the following line to Program Listing 2: 140 X=RND(X).

For scientific studies, always use a pseudo random number sequence which you can repeat if called on to do so.

There you have it: the connection between memory, plates, and random numbers. Next time you watch your computer spin out fantastic results on the video screen, reflect for a moment on the amazing Z-80 juggler in the heart of your computer keeping all those plates spinning, balls bouncing, rings twirling, chairs balanced, while doing a handstand on a tightrope high above... ■

```
100 DEFINT A-Z: DIM A(127)
110 L=16555
120 FOR I=1 TO 1000
130 RANDOM:REM SET SEED TO REFRESH COUNTER
140 X=PEEK(L)
150 A(X)=A(X)+1
160 NEXT I
170 CLS:REM PLOT FREQUENCY OF OCCURENCES
180 FOR I=0 TO 127
190 Y=A(I):IF Y>47 THEN Y=47
200 FOR J=0 TO Y:SET (I,47-J):NEXT J
210 NEXT I
220 GOTO 220
```

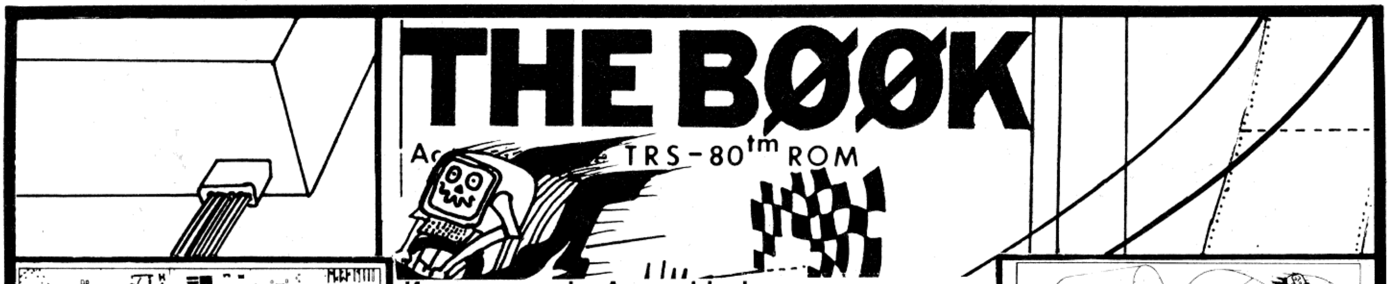
Program Listing 1

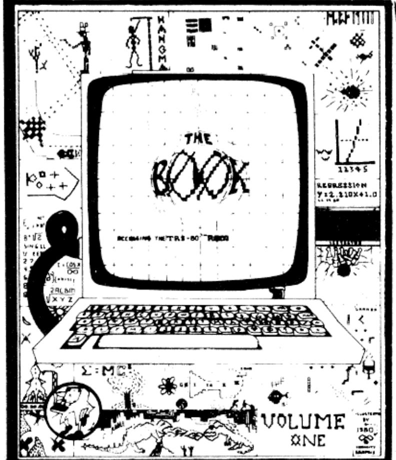
```
100 DEFINT A-Z: DIM A(127)
110 L=16555
120 FOR I=1 TO 1000
130 X=RND(127)
150 A(X)=A(X)+1
160 NEXT I
170 CLS:
180 FOR I=0 TO 127
190 Y=A(I):IF Y>47 THEN Y=47
200 FOR J=0 TO Y:SET (I,47-J):NEXT J
210 NEXT I
220 GOTO 220
```

Program Listing 2

✓ Reader Service—see page 242


Taranto
& ASSOCIATES, INC.
The Total System Store





If you ever do Assembly language programming, or you just want to know more about your TRS-80 ROM, "THE BOOK"s are for you.

*TRS-80 is a trademark of Tandy Corp.

Each volume is priced at: $14.95 + $1.50 S&H = $16.45 ($17.05 in VA) Overseas add $2.50 for air shipment

Insiders Software Consultants, Inc. ✓305
P.O. Box 2441, Dept. S. M2
Springfield, VA 22152

☐ **Please send me Volume I of THE BOOK**

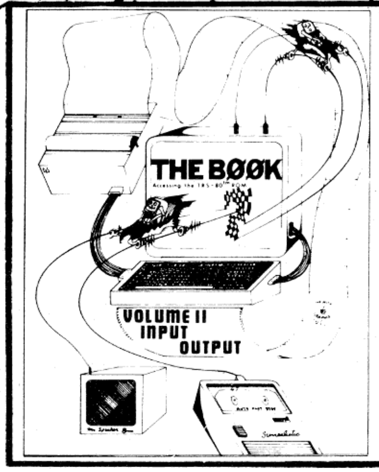
☐ **Please send me Volume II of THE BOOK**

NAME ______
ADDRESS ______
CITY, STATE ______ ZIP CODE ______

☐ Check payable to Insiders Software Consultants, Inc.
☐ MASTER CHARGE MC Bank Code
☐ VISA Exp. Date ______ Card Number ______

Signature ______

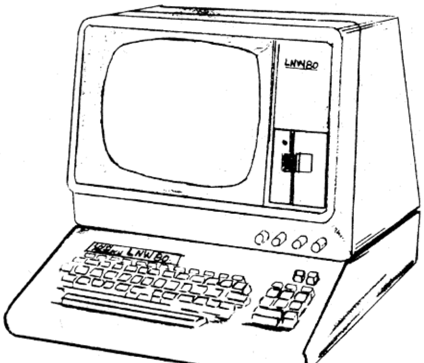
Dealer Inquiries Invited



Volume I: The most comprehensive book yet on the math routines in Level II ROM, Models I & III. Includes a fully commented listing from 0708H to 1607H and an incredibly complete map of the ROM and reserved areas of RAM.

Volume II: Everything you want to know about the video, keyboard, cassette, and printer driver routines. Learn how to write your own! Remarkably detailed listings illustrate well-commented source code. Complement Volume I, now.

THE FIRST TRS-80® COMPATIBLE COMPUTER WITH HIGH DENSITY COLOR GRAPHICS!



LNW RESEARCH introduces the LNW80, a high performance color computer, compatible with the TRS-80™ Model I. The fully integrated LNW80 is a sophisticated and versatile microcomputer with the following powerful features.

COMPATIBILITY
Hardware and software compatible to the Radio Shack TRS-80™ Model I computer, provides the widest software base of any microcomputer. Cassette interface; expansion bus.

DISPLAY
Quality upper and lower case display.

Two modes of color graphics, high resolution graphics, 384 × 192 in eight colors—higher density than the Apple II.* Low density color graphics of 128 × 192 are also available in eight colors.

High resolution—black and white graphics—of 384 × 192 mixed with text and TRS-80™ standard graphics.

Reverse video, composite video RF output.

PERFORMANCE
The LNW80 utilizes the fast Z-80A microprocessor which executes at a speed of 4 MHZ—over twice the speed of the TRS-80™ Model I.

LNW80

PC BOARD $89.95

Ask about our : Keyboard cabinet, Leedex VIDEO 100-80

NEW EXTERNAL DATA SEPARATOR
ASSEMBLED AND FULLY TESTED
$14.95
SOME SOLDERING REQUIRED

SYSTEM EXPANSION
AT $69.95 [PC BOARD & USER MANUAL]

- SERIAL RS232C/20 mA I/O
- FLOPPY CONTROLLER
- 32K BYTES MEMORY
- PARALLEL PRINTER PORT
- DUAL CASSETTE PORT
- REAL-TIME CLOCK
- SCREEN PRINTER BUS
- ONBOARD POWER SUPPLY
- SOFTWARE COMPATIBLE
- SOLDER MASK, SILK SCREEN

LNW RESEARCH
✓53

LNW RESEARCH 3183-E AIRWAY AVE COSTA MESA CA 92626 714-641-8850

*Apple II is a TM of Apple Computer, Inc. TRS-80 is a TM of Tandy Corp.

ORDERING INFORMATION
Add $3 for postage and handling. CA residents add 6% sales tax

VISA | master charge

This project will keep you busy for awhile.

A Very Versatile Interface

W. R. Stanley N4TF
204 Avery Lane
LaGrange, GA 30240

Why is an interface usually required for the TRS-80 or for almost any other computer? What characteristics do most interfaces possess?

An Example

In our analysis of interfaces, let's use the Model 15 TTY printer as a good example. No, this won't turn into just another Model 15 article.

The Model 15 selector magnets require a predetermined sequence of 60-ma current pulses, Each is about 22 ms in duration to allow the machine to print a character or perform a function. The TRS-80 cannot directly drive those selector magnets.

First, the output current capability of the internal circuits connected to pins on the keyboard connector is measured in either the microamp or low milliamp range.

Second, the information present on the computer data bus has durations measured in the nanosecond or microsecond range.

Third, the CPU uses the data bus at the connector for more than outputting data to an external circuit or device.

The TRS-80 is simply not designed to operate external devices that require large voltage swings or high current levels.

The interface inserted between the keyboard and the Model 15 selector magnets must, therefore, contain circuits that will translate feeble signals from the computer to the power level to run the magnets.

The Z-80 CPU in the TRS-80 operates according to instructions and applied data. Its operations are synchronized by clock pulses applied to that chip: the total duration of one clock pulse is about 560 nanoseconds. Since most operations occur in a time period of a few clock cycles, the TRS-80 cannot produce data pulses measured in the millisecond range.

"The TRS-80 is . . . not designed to operate external devices that require large voltage swings or high current levels."

If the external device requires operating signals exceeding a microsecond, our interface must produce longer pulses. Again, using the Model 15 as an example, a 22-ms pulse could be generated by the interface in the following way.

Program the CPU to output a data pulse that causes a circuit in the interface to turn on. That circuit will cause the output of the interface to turn on. Next instruct the CPU to perform a series of instructions, either useful or nonsense, that will take a total time of about 22 ms. Next, tell the computer to output another data pulse that causes the interface output signal to end.

An alternate way to produce the desired pulse length is use of a one-shot multivibrator in the interface. This performs the timing function. The computer still provides an output signal denoting the starting time of the desired pulse. It's also necessary not to send pertinent data to the interface until the external timing device ends its current cycle. Timing of the CPU operations is still of significance.

The next consideration is that the data bus at the keyboard connector is connected to several discrete memory blocks inside the computer. This bus is used at times in the operations of the CPU, i.e., for fetching an instruction from dynamic memory, writing data to video memory, or looking for an input from the keyboard.

You can now see that the interface is required to isolate the computer from the outside world. Otherwise, the CPU might misinterpret some electrical characteristic of the external device into an instruction or data word. This means that the computer ceases functioning in the proper manner.

The Requirements

As you read this article, or others devoted to interfacing, keep in mind the basic requirements of interfaces and see how they are met. The requirements are: produce signals of voltage or current levels to drive the external device (or convert external signals to levels suitable for application to the computer circuits); modify signal waveform or duration; and isolate the computer from the external device until the CPU is instructed to communicate with it.

Circuit

The circuit diagram of a versatile interface board is shown in Fig. 1. The capabilities of this board are obvious as you consider the characteristics of the 8255 chip, the heart of this circuit.

The 8255 Programmable Peripheral Interface was originally manufactured as a part of the 8080 CPU family. Since the Z-80 CPU used in the TRS-80 includes the 8080 instruction set, the full range of operational configurations of the 8255 are available to the user.

The 8255 is software that can be programmed by writing specific data words (called control words) to an on-chip register at a certain port address. It is possible through proper software control to change the operating characteristics of the 8255 at any desired point to suit the situation. For example, in one section of a program the 8255 could be shaped to store data from an outside source and read into the computer at a certain time. The characteristics of the 8255 could be reprogrammed (within the same program) to output CPU data for use in some external circuit.

The basic configuration is a group of 24 I/O pins divided by software control into two eight-bit and two four-bit ports. These are ports A, B, C low and C high. In the operating mode of the interface board, port 00 is the address of port A, port address 01 represents port B, and port address 02 represents port C low and port C high.

The 8255 control register is accessed by writing to port 03. Anytime the shape of the chip is to be changed, the appropriate control word is sent to that port address. Reading of data stored in the control word register is not provided by the manufacturer.

Note on the schematic that the RESET line from the TRS-80 is inverted and applied to the 8255. The RESET pin on this chip is low for normal operation. When you press the RESET button on the keyboard, the 8255 configuration will automatically be changed to where three eight-bit ports on the chip are in the input mode.

As said before, the 8255 can be programmed to perform many functions. Each is designed to properly condition some characteristic of the transferred data signals in a particular direction between the CPU and some external device. The 8255 acts as an isolator between the computer and that device, to prevent interference between the two.

A few of the many operating configurations of the 8255 will be covered. I suggest that you get the manufacturer's data sheet (actually a booklet) when you acquire the chip. Another source of information is the *Intel Component Data Catalog.* This publication contains complete data on the 8255, and a wealth of information on many other chips of interest to the breadboarding computer hobbyist.

Now to the actual interface circuit. Keep in mind that this is a general-purpose interface, useable by itself or with additional circuitry. These properly interface a variety of devices with the TRS-80.

Address lines A_0 and A_1 are routed from the keyboard to the 8255 address inputs. These two lines are adequate for addressing the four ports in our application. Address lines A_2 through A_7 are applied to an address decoder circuit.

The address decoder functions along with the control logic circuit to enable the interface board. This is done only when data is transferred between the CPU and the four ports on the board. The interface is transparent to the CPU for all other port addresses and memory-mapped locations.

We want the 8255 active only when the TRS-80 is instructed to communicate with that chip; therefore, the $\overline{IN}$ or $\overline{OUT}$ signal at the keyboard connector is combined with the address decoder output. If a port address in the proper range appears along with either the $\overline{IN}$ or $\overline{OUT}$ control signal, an enabling voltage is applied to the $\overline{CS}$ pin on the 8255.

Note that the $\overline{IN}$ or $\overline{OUT}$ control signal, after being combined with a true output from the address decoder, is applied to the $\overline{RD}$ or $\overline{WR}$ pin, respectively, on the 8255. This chip, therefore, receives the control signals only under appropriate conditions.

If the address decoder senses any port address above 03, its output will not permit the application of any control signals to the interface. The interface circuit will then appear to the TRS-80 to be disconnected.

Note on the schematic the presence of two sets of three-state data bus buffers. One set operates when data is written to the interface, while the other is used when the direction of data flow is reversed. Each set receives an enabling signal only when the address decoder senses the appropriate address range, and an $\overline{IN}$ or $\overline{OUT}$ control signal is present. At all other times both buffer outputs will disconnect the interface from the data bus.

The astute reader might

wonder why data bus buffers are employed in this circuit, keeping in mind that the 8255 has a three-state data bus buffer capability already built in. My experiments reveal that external data bus buffering is sometimes necessary due to relaxed parameters of chips available at bargain prices. My advice is to utilize the external three-state buffers as added insurance against difficulties you might encounter.

Construction

Component layout is completely noncritical; however, you might spend a few moments trying different IC arrangements to make wiring fairly simple. I urge you to use sockets for all ICs, because it will make component replacement easier.

I built the circuit shown in Fig. 1 on a Radio Shack experimental breadboard. My construction is centered on use of this board because it provides ample room for wiring fairly complex circuits.

If you are following in my footsteps, obtain two breadboards and two mating connectors for those boards. One will be used to terminate one end of the cable between the interface board and the keyboard. The other 44-pin connector is epoxied to the top of the interface board. It serves as a convenient means of connecting the interface board to additional circuits.

The second board will be used in construction of a circuit that I feel many readers may find usable.

The connector that goes with the keyboard output is a 40-pin connector with 0.1-inch contact spacing. It is available from several companies advertising in this magazine and similar periodicals.

The interconnecting cable, with a recommended maximum length of about one foot, is terminated on one end with the 44-pin connector. Of course, the 40-pin connector is used on the other end. You can use ribbon cable or individual stranded wires to form the conductors.

As you fabricate this cable, you could provide only the necessary connections between the TRS-80 and the interface board. You can also look to future needs and connect all 40 corresponding pins. The latter approach would probably cut down on cable-making later.

I suggest using the same pin assignments for both connectors (pin 1 to 1, 2 to 2, etc.). If you use this method, other 44-pin boards can be connected to the TRS-80 using the same cable.

What about the four extra pins on the 44-pin connector? These are to be used as you see fit, to connect the outputs of one or more external power supplies to the board affixed to that connector.

Don't forget the ground connection between the external supplies and the connector. Use any pin designation that suits your fancy, but have it well-documented. Remember, you might want to use this cable and the same power supplies on another board.

Caution

One word of caution regarding the 40-pin connector. Depending on the manufacturer, the pin number designations might not agree with the pin as-

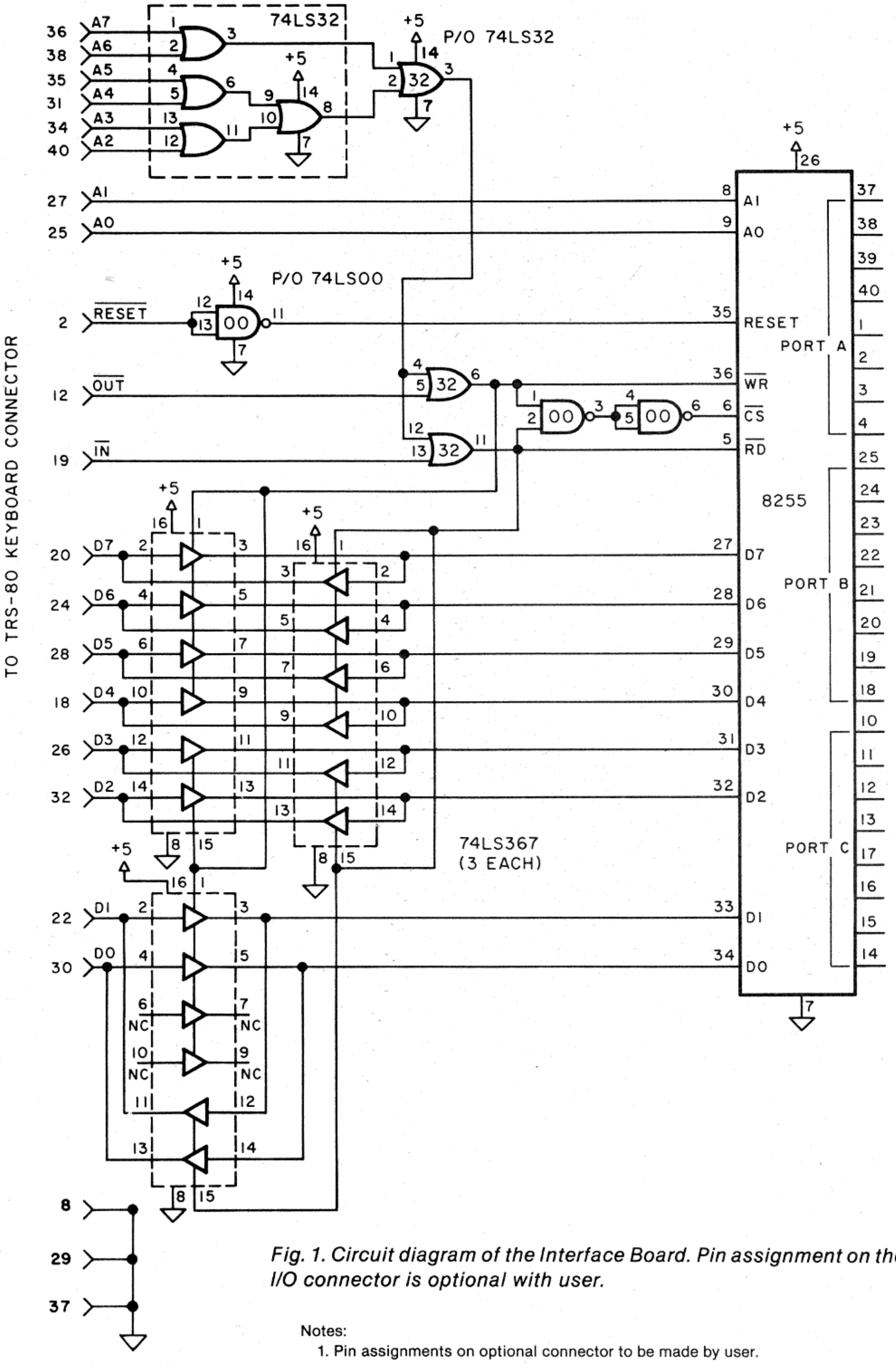


Fig. 1. Circuit diagram of the Interface Board. Pin assignment on the I/O connector is optional with user.

Notes:
1. Pin assignments on optional connector to be made by user.
2. Adjust value of R_1 as necessary (± 50 percent) to obtain proper operation with selected transistor.
3. Make no substitutions for the 7407. Maximum voltage ratings of other open-collector inverting buffer ICs might be exceeded during programming.

PENCIL-FIX

Save your warranty. Use PENCIL-FIX to avoid custom control key on your keyboard. Redefines the control key for EP to be the @ key. Use RS lower case mod or the EP mod without the control key. Disk based EP only. . . **$14.95**

PRINT-CENTRAL

Send any control code directly to your smart printer from the BASIC command mode or from DOS. Avoid having to type such things as "LPRINT CHR$(31) just to change pitch. Use CLEAR right arrow (two key strokes) instead. Any code from 1 to 31 may be sent. Mod I 32K disk. . . **$24.95**

SPOOL-REL

An in-memory print buffer that runs in Mod I 32K or 48K disk systems. Fully relocatable code and buffer. A true background spooler at an unbelievably low price. . . **$24.95**

TIGGER-GRAF

Create engineering, scientific, business, or just plain fun on your IDS 440G or 460G printer. Resolution is 495 x 575. Easy BASIC programs provided for data entry and machine language module for speed. Includes setting individual points, drawing lines, shading shapes. Several graphs may be catenated along the Y-axis for larger graphs. Requires Mod I 32 or 48K 1 disk. . . **$149.95**

THE COMPUTER STORE

4949 SOUTH PEORIA AVE.
TULSA, OKLA 74105
918-747-2550

DOES YOUR SMART PRINTER SUDDENLY BECOME DUMB WHEN YOU PRINT FROM SCRIPSIT?

Is your printer capable of underlining
but not from Scripsit?

Is your printer capable of **Bold Printing**
but not from Scripsit?

Can your printer super script and sub script
but not from Scripsit?

Can your printer change pitches
but not from Scripsit?

THE ANSWER IS SCRIPMOD

SCRIPMOD does not require a separate printer driver. With SCRIPMOD control codes can be embedded in the text of your document. You use the same format line sytax you're used to now. SCRIPMOD adds one format instruction to Scripsit & two control codes.

>C = N J = N CC = H

The above line tells SCRIPMOD to justify text and to backspace the printer upon encountering a Text Character in the document. The Text Character is entered by placing the cursor at the place in text where the control code is to take place and typing an @T. Of course your printer must be capable of backspacing and must use the ASCII code 8 (or H) as the control for backspace. If the code is something different there is no problem as the correct code may always be sent. Any control code your printer is capable of using from 1 to 31 with or without an ESCAPE lead in may be sent.

The second control code which is added is the MENU command. You press an @M and the screen clears and prompts you to select a drive from 0 through 3 or return to the text. All visible files can be displayed on the screen at this time. When you select to return to the text, the cursor is placed on the exact character it was on when you selected @M.

This ad was written with SCRIPMOD and the Daisey Wheel II printer.

The minimum system required for SCRIPMOD is the Mod I 32K disk system with either the RS lower case mod or the EP lower case mod.

SCRIPMOD is supplied on disk with full documentation for $39.95.

VERN STREET PRODUCTS ✓432
114 West Taft
Sapulpa, Oklahoma 74066
Phone: (918) 224-5347

HI-RESOLUTION GRAPHICS FOR TRS-80*

INTRODUCING:

E/RAM

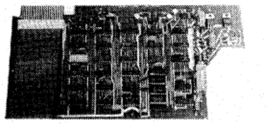
E/RAM Graphics is a unique hardware/software package, which will integrate high-speed, high resolution graphics into any Level II TRS-80 system. E/RAM hardware is a fully plug-compatible box, which installs in minutes, and requires absolutely **no** modifications to the TRS-80 system. E/RAM software is a compact, relocatable set of utilities which provides the user with easily accessible graphics functions. For instance: the user pokes the end point coordinates of a line into certain locations, does a USR call, and an optimized dot-raster line is automatically drawn on the screen at very high speed (less than 10 milli-seconds for a medium length line).

E/RAM does not require the purchase of an additional monitor CRT. The high-resolution graphics video is syncronized with the TRS-80 video and appears on the screen with the normal TRS-80 display. Alphanumerics, TRS-80 graphics, and E/RAM high-resolution graphics may be displayed simultaneously or individually.

E/RAM hardware contains its own 6144 byte video memory, which provides a true 256 x 192 matrix of **independent** graphic elements. (E/RAM is **NOT** a programmable character generator type graphics system. Character generator systems have serious limitations in full screen graphics applications.)

E/RAM will operate with or without an expansion interface, and with any standard memory configuration (4k through 48k).

E/RAM is **fast**. "E/RAM" is an acronym for **E**xtended **R**andom **A**ccess **M**emory, a very short description of the Patent-Pending method of I/O employed by this device, which gives it memory-mapped speed without interfering with the memory space used by the TRS-80.



The installation of E/RAM will not affect normal operation of the TRS-80. High resolution ON/OFF is under program or manual control (a switch is provided). An expansion card edge connector is provided so that other peripherals may be used on the TRS-80 bus.

E/RAM software package is compact (less than 1000 bytes), fast, easy to use, and very flexible. A relocating loader is provided. The user can delete unneeded routines if more memory space is required. Lines can be drawn as fast as 13 per second using BASIC USR calls, and as fast as 200 per second using assembly language programs.

Routines usable through USR of BASIC, and of course an assembler CALL are:

INIT	- Sets up display
PLOT	- Plots a point
READ	- Reads a point from the screen
BLACK	- Sets drawing mode to black (off)
WHITE	- Sets drawing mode to on
CLEAR	- Clears the high-resolution graphics screen
LINE	- Draws a line

As an example, after the utilities package is loaded and you desire to draw a line, the following sequence of BASIC instructions could be executed:

U=USR(0)	Return the communications area
POKE U+1,X0	Provide the beginning X coordinate
POKE U+3,Y0	Provide the beginning Y coordinate
POKE U+5,X1	Provide the ending X coordinate
POKE U+7,Y1	Provide the ending Y coordinate
V=USR(4)	Draw the line (Current speed is approximately 13 vectors/second)

The complete E/RAM package is available for only $349.95, and includes case, power supply, cables, software cassette, and complete documentation.

To order, or for further details, write or call:

VERN STREET PRODUCTS ✓432
114 West Taft
Sapulpa, Oklahoma 74066
Phone: (918) 224-5347

We handle a full line
of Radio Shack products

Send $10.00 for a set of the manuals provided (applicable towards purchase)

Dealer inquiries are invited.

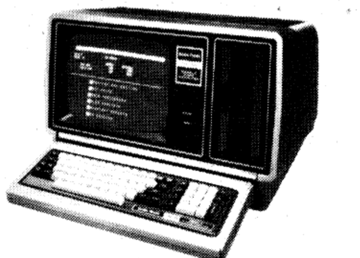
Terms: COD Welcome, check, money order, Master Charge, or Visa

Delivery: Stock to 60 days.

E/RAM was designed, and is manufactured by KEYLINE COMPUTER PRODUCTS, INC. 13 East 6th Street, M/C 200, Tulsa, Oklahoma 74119.

*TRS-80 is a registered trademark of Radio Shack, a Tandy Corporation.

If you're looking for the best prices in the U.S.A. on



TRS-80®

MICROCOMPUTERS

We have consistently offered the TRS-80 line at savings **up to 20%,** which means you can save $150 to $1500 by buying directly from Computer Discount of America.

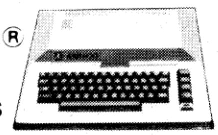
TRS-80 Model II, 64K System, with disc drive only $3385.00

Other TRS-80 Model II, or Model III computers and systems, Color Computers, and Pocket Computers are in stock at similar savings.

Our savings are as big on expansion interfaces, printers, diskettes – everything for your TRS-80 System.

ATARI®

MICROCOMPUTERS



We have the full line of ATARI personal computers and systems, including Models 400 and 800. The computers, accessories, and hardware are brand new, in factory sealed cartons, and carry a full factory warranty. Most models are in stock for immediate delivery (usually within 7-10 days), and a price quote is as near as your phone. So if you're looking for the **best prices in the U.S.A.,** for microcomputers and accessories, call Computer Discount of America, Inc., West Milford, New Jersey 07480. 201-728-8080. **NO TAX ON OUT-OF-STATE SHIPMENTS.**

TOLL FREE 800-526-5313

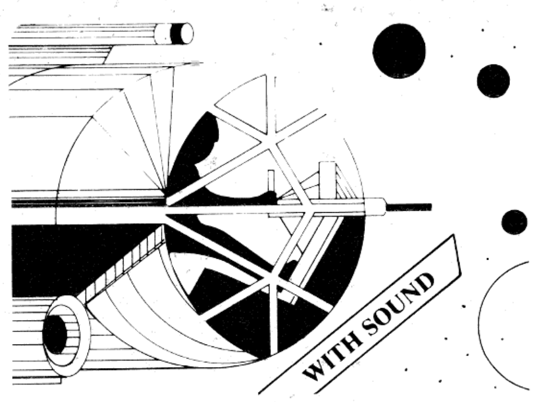
✓372

Computer Discount of America ®

Authorized TRS-80 dealer, store B-282.

Go Boldly

Where No TRS-80 Program Has Gone Before!



ASTEROID

DATE: 28.02.2047
LOCATION: 270 million miles from Terra
MISSION: Maintaining Terra's Space Lanes

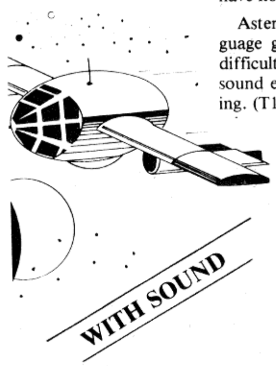
Briefing will follow:
1.1 Your mission is to destroy any asteroids in your sector and to prevent alien spacecraft from infiltrating the Terran Defense Network.
1.2 Your ship is armed with an anti-matter cannon. You can shoot large asteroids, but this turns them into many smaller asteroids, each capable of destroying your ship.
1.3 In addition, alien ships can make instantaneous hyperspace jumps into your area and start firing on your ship.
1.4 You'll need lightning reflexes and nerves of steel to survive Asteroid. We have no use for non-survivors!

Asteroid, a real-time, machine-language game, features variable levels of difficulty, superb high-speed graphics, sound effects and automatic score keeping. (T1) or (T2)

Order No. 0237R
$14.95. Tape
Order No. 0247RD
$19.95. Disk

COSMIC PATROL

WARNING: PLAYERS OF THIS GAME SHOULD BE PREPARED FOR A STATE OF REALISM HITHERTO UNAVAILABLE ON THE TRS-80



Skilled players soon master many difficult computer games, but COSMIC PATROL is in a world all its own. The challenge intensifies! Supporting graphics and sound (optional) make each encounter an exciting new experience. It all adds up to a Super 3-S package...skill, sight and sound.

Scenario: The Cosmic Patrol program puts you in the command chair of a small interstellar patrol craft. Your mission is to defend Terran space and prey on the Quelon supply ships which carry essential parts and lubricants for that implacably hostile robotic force. The drone freighters are fairly easy pickings for the accomplished starship pilot, but beware of the I-Fighter escorts. They're armed, fast and piloted by intelligent robots linked to battle computers. They *never* miss.

The Cosmic Patrol program is not just another search and destroy game. With its fast, real-time action, impressive sound option and superb graphics, this machine-language program is the best of its genre.

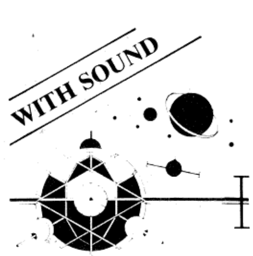
Don't keep putting quarter after quarter into arcade games or spending big bucks for video game cartridges. Get Cosmic Patrol from Instant Software—and get the best for less! (T1)

Order No. 0223R
$14.95 Tape.
Order No. 0224RD
$14.95 Disk.

BALL TURRET GUNNER

For years the Petro Resource Conglomerate has attacked our photon collection stations and strangled our deep-space trade routes. The PRC Exxonerator Class light fighters (code name: Gnat) have been their main weapon. Now you can strike back, by joining the Ball Turret Gunner Service.

Imagine yourself at the control console of an LW-1417 Stratoblazer (Type B Strategic Laser Weapon). Your Hindsight Director informs you that a Gnat fighter is coming in for an attack. You pivot your gigawatt laser turret until you can see the target on your monitor. The Range Indicator shows him coming in fast. The Targeting Computer studies his course and speed as your finger tenses over the firing key. You know you'll have only a fraction of a second in which to react. The Gnat fighter's evasive maneuvers cause him to dance in your sights. Suddenly,



you see the FIRE Command and you react instinctively. Your laser beam lashes out and reduces the Gnat to an expanding ball of ionized gas. Mission accomplished!

Ball Turret Gunner, with your choice of multiple levels of difficulty, optional sound effects and superb graphics, is more than just a game. It's an adventure. Experience it! (T1)

Order No. 0051R $9.95.

✓2

Instant Software™

PETERBOROUGH, N.H. 03458 603-924-7296

TO ORDER
See Your Local Instant Software Dealer or Call Toll-Free 1-800-258-5473

*A trademark of Tandy Corporation

signments on the male connector at the rear of the keyboard. On the keyboard connector, all odd-numbered pins are on top of the circuit board, with pin one farthest from the RESET button. Pin two is directly under pin one. Pin 39 is directly above pin 40, which is closest to RESET.

Use some suitable marking medium on th 44-pin connector and all boards to be inserted into it, to insure correct connections. A reversal at either end of the cable would most likely disturb the operations of the TRS-80.

A simple power supply furnishing regulated +5 V at 0.25 amp would be more than adequate for powering the interface board.

Applications

If you have a specific use for the interface board, use the data sheet for the 8255 to determine the control word(s) that will form the chip for your applications. Instruct the computer to write the desired data to the control word register. This will give the interface board the input and/or output capabilities needed for the application.

I use the interface board in conjunction with another circuit board to program 2708 EPROM. This second board is inserted into the connector mounted on top of the interface board. It contains all the additional circuits necessary for programming and verifying the 2708.

The interface board is used during programming for latching output data from the TRS-80, to be presented to the 2708. When data is read from the 2708 in preparation for verifying its stored data, the interface board will present latched computer output data to the EPROM and transfer to the computer the data read from the 2708.

Programming the 2708

The act of programming consists of writing eight-bit hex bytes to discrete memory locations. Since the memory cells in an erased or new chip contained binary ones, only the zero bits in the bytes being programmed into the chip actually cause changes to take place in the chip. The only way a zero bit can be changed to a one is to erase the EPROM and reprogram it.

Programming must be done in a definite sequence, with timing restrictions placed on certain signals and voltages applied to the 2708. For maximum life of the programmed data, each memory address on the 2708 should be programmed for a total time span of about 100 milliseconds. The maximum time allowed for programming a location is roughly one millisecond; therefore, if each location is programmed for one ms at a time, approximately 100 complete programming cycles, or loops, would be indicated.

All addresses on the chip must be stepped through in sequence during programming. The manufacturer does not advise programming only selected portions of the 2708 memory, as can be done with some other PROMs.

The appropriate address data is presented to the 2708, followed by the data byte to be programmed into that location. After a short delay, to allow those two inputs to become stable, a pulse of about 26 volts at one ms is applied to a programming pin on the 2708. If the voltage on the $\overline{CS}$/WE pin on the chip had been raised to +12 V, the programming pulse causes the data presented at the indicated address to be stored into the EPROM.

At the end of this programming pulse, the address data presented to the 2708 is incremented to the next location. Data for that location is then applied to the chip, followed by another programming pulse. This process is repeated for the 1024 locations on the chip. The above is then repeated 100 times.

After completing programming of the 2708, the voltage applied to the $\overline{CS}$/WE pin is decreased to either +5 v or 0 v. The five-volt level disables the chip, while zero volts allows data to be read from the chip when other conditions are met.

The result of the above should be a properly programmed 2708, provided the data entered was correct. Cover the quartz window with an opaque material

✓Reader Service—see page 242

such as black electrical tape to prevent gradual erasure by exposure to light over an extended period of time.

Remember that address and data information must be applied to the 2708 before, during and shortly after the one-ms programming pulse is applied to the chip. The latching output functions on the 8255 board are ideal for this application, and the interface board is used to control the circuit that provides the programming pulse.

EPROM Burner Construction

Fig. 2. contains the schematic of the 2708 EPROM programmer board. This board is designed to be plugged into the 44-pin connector glued to the top of the interface board. This is why the purchase of two boards and two connectors was recommended.

However, those interested in a combination of the interface board and the 2708 programmer circuit should have no difficulty in building the entire circuit on one board.

A standard 24-pin socket for the 2708 will be adequate for occasional use of the EPROM programmer. For continual use, the additional cost of a zero insertion force socket would be worthwhile.

There are no tricks involved in construction of the programmer circuit board. You will have to provide a source of +26 v regulated at about 20 ma, in addition to the conventional +12 v, +5 v and −5 v supplies. For the three latter voltages, I used a dual-secondary transformer and three separate rectifier-filter-regulator circuits. Each regulated output has a one-amp current capability, far more than is needed. Yet, it leaves open future options if you want to add onto those power supplies later.

The +26 v regulated supply is also straightforward. I used another rectifier—filter circuit to provide about 30 v to a LM317 adjustable regulator. This IC is trimmed to provide the proper voltage at its output.

The 26 v lead from the regulator is connected to the EPROM programmer circuit by a plug and jack. Since voltage and current ratings of the connectors are of no great concern, you might use subminiature audio system components. The plug is connected to the programmer circuit by a short flexible lead. No connection is made to the shell, or outer part of either of the mating connectors.

The way the +26 v source was connected to the programmer circuit is used for a good reason. It isolates this voltage level from any other circuits even remotely connected to the computer. The separate connection for the +26 v supply serves to heighten caution to the user in its application.

Software

Program Listing 1 contains a hex dump of my version of a software driver for the 2708 programmer circuit. It is operated with the 8255 interface board.

The execution address of this program is 7000 hex. Its location in this area puts it out of the way of other monitor programs you might use in conjunction with this program. The driver uses the entire memory block between 6000 and 67FF at different points in its operations, so keep that area clear of extra material.

The program now contains several absolute addresses that would prevent block-transferring it to some other address range without internal modification. Relocation and address-changing by another program might not result in proper operation at the new location, because data storage locations and ASCII message characters appear in the program.

Type this program into memory, using a monitor such as T-BUG. Punch at least one copy, to save a lot of typing later on.

After you get the program loaded into the computer, run it with power removed from the programmer circuit. This will familiarize you with the actual programming process.

You will see a display on the screen which leads you through the operation. The EPROM program uses the memory block 6000-63FF for storage of the actual program to be placed in the 2708. It uses locations 6400-67FF for reading of the

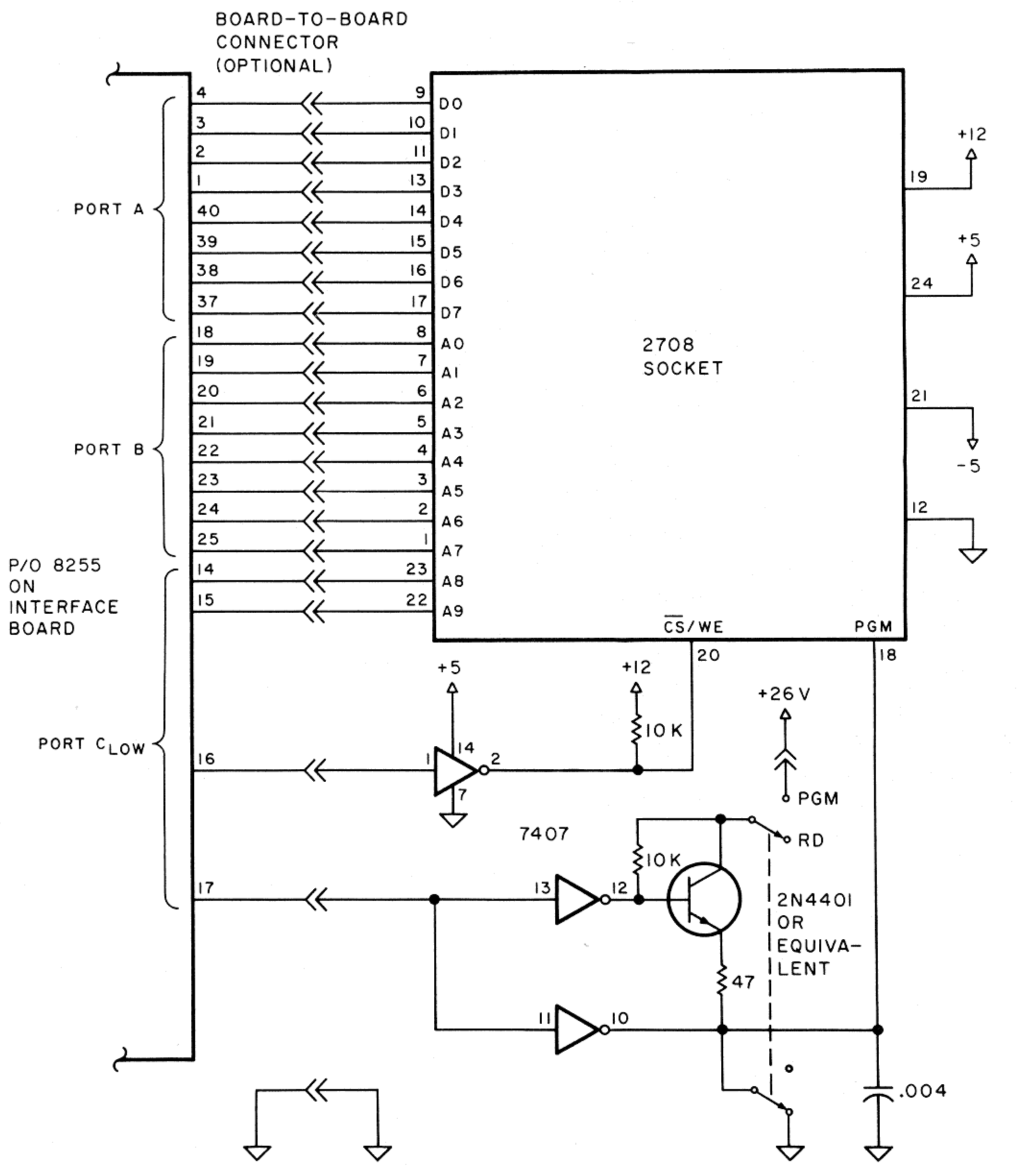


Fig. 2. 2708 EPROM Programmer circuit diagram. This circuit may be built on the Interface Board, or built on a separate Radio Shack card.

EPROM contents back into computer memory. This is used for later comparison with the program starting at 6000.

In the verify mode, the program compares the contents of sequential locations in those memory blocks. It prints on the screen the hex address where the locations differ in data content.

If you wish to make sure that the 2708 is erased, you can use the load and verify functions to do so. Write a simple program (possibly using the LDIR instruction) to load FF into all locations in the memory block 6000-63FF.

Insert the 2708 in its socket and apply power to the interface and programmer circuts. Check to see if the switch on the EPROM board is in the READ position. Enter the appropriate command to load the 2708 contents into memory. Then use the verify function to see if all 2708 locations contain FF.

If any locations in the 2708 contain some value other than FF, that chip is not erased. If a portion of the 2708 has already been programmed, use your monitor program to look at the addresses in the 6400-67FF block. Satisfy yourself that the program already in the EPROM is appropriate in location and content.

If everything checks out, you are ready to begin the programming operation. Using your monitor, exercise great care in entering the data to be burned into the 2708, and enter this data in the one k block starting at 6000.

If all the memory of the 2708 is not to be written into, make sure that all locations in the 6000-63FF block, with the exception of the portion containing the program to be burned in, contain FF. This way, you can add short program segments later on into unused portions of the 2708.

After you are satisfied that the data entered into the 6000 block is correct, reenter the EPROM program and invoke the PROGRAM command. Follow the instructions in setting the read-write switch. The programming operation, covering 100 loops of programming for all locations on the 2708, will take a little more than two minutes.

```
7ØØØ  3E 1C CD 33 ØØ 3E 1F CD 33 ØØ 21 8E 71 CD A7 28
7Ø1Ø  CD 84 Ø3 CD 33 ØØ FE 5Ø CA 2A 7Ø FE 4C CA 92 7Ø
7Ø2Ø  FE 56 CA B7 7Ø FE 54 CA AØ 73 21 ØD 72 CD A7 28
7Ø3Ø  CD 2B ØØ FE Ø1 CA ØØ 7Ø FE ØD 2Ø F4 C3 23 71 ØØ
7Ø4Ø  3E 8Ø D3 Ø3 3E 64 F5 21 ØØ 6Ø 11 ØØ Ø4 7C E6 Ø3
7Ø5Ø  F6 Ø4 D3 Ø2 F5 7D D3 Ø1 7E D3 ØØ Ø6 1Ø 1Ø FE F1
7Ø6Ø  F5 F6 Ø8 D3 Ø2 D9 21 ØØ FØ 11 ØØ FØ Ø1 58 ØØ ED
7Ø7Ø  BØ D9 F1 D3 Ø2 23 1B 7A FE ØØ 2Ø Ø3 7B FE ØØ C2
7Ø8Ø  4D 7Ø F1 3D F5 FE ØØ C2 47 7Ø 3E 9B D3 Ø3 F1 C3
7Ø9Ø  36 71 3E 9Ø D3 Ø3 21 ØØ 64 11 ØØ Ø4 7C E6 Ø3 D3
7ØAØ  Ø2 7D D3 Ø1 DB ØØ 77 23 1B 7A FE ØØ 2Ø Ø3 7B FE
7ØBØ  ØØ C2 9C 7Ø C3 ØØ 7Ø 21 ØØ 6Ø 11 ØØ 64 Ø1 ØØ Ø4
7ØCØ  1A BE CA CA 7Ø C5 CD DB 7Ø C1 23 13 ØB 78 FE ØØ
7ØDØ  2Ø Ø3 79 FE ØØ C2 CØ 7Ø C3 13 71 7C CD E9 7Ø 7D
7ØEØ  CD E9 7Ø 3E 2Ø CD Ø9 71 C9 4F CD EE 7Ø C9 CB 3F
7ØFØ  CB 3F CB 3F CB 3F CD FC 7Ø 79 E6 ØF C6 3Ø FE 3A

71ØØ  FA Ø5 71 C6 Ø7 CD Ø9 71 C9 D5 FD E5 CD 33 ØØ FD
711Ø  E1 D1 C9 21 36 72 CD A7 28 CD 2B ØØ FE ØD 2Ø F9
712Ø  C3 ØØ 7Ø 3E 1C CD 33 ØØ 3E 1F CD 33 ØØ 21 6Ø 72
713Ø  CD A7 28 C3 4Ø 7Ø 21 49 71 CD A7 28 CD 2B ØØ FE
714Ø  ØD CA ØØ 7Ø 2Ø F6 C3 ØØ 7Ø ØD ØD ØD ØD 5Ø 52 4F
715Ø  47 52 41 4D 4D 49 4E 47 2Ø 43 4F 4D 5Ø 4C 45 54
716Ø  45 44 ØD ØD 53 45 54 2Ø 53 57 49 54 43 48 2Ø 54
717Ø  4F 2Ø 52 45 41 44 ØD ØD 48 49 54 2Ø 45 4E 54 45
7188  52 2Ø 54 4F 2Ø 43 4F 4E 54 49 4E 55 45 ØØ ØD 45
719Ø  5Ø 52 4F 4D 2Ø 5Ø 52 4F 47 52 41 4D 4D 45 52 ØD
71AØ  ØD 5Ø 52 4F 47 52 41 4D 2Ø 54 4F 2Ø 42 45 2Ø 42
71BØ  55 52 4E 45 44 2Ø 4C 4F 41 44 45 44 2Ø 41 54 2Ø
71CØ  36 3Ø 3Ø 3Ø 48 ØD ØD 5Ø 2Ø 3D 2Ø 5Ø 52 4F 47 52
71DØ  41 4D ØD 4C 2Ø 3D 2Ø 4C 4F 41 44 ØD 56 2Ø 3D 2Ø
71EØ  56 45 52 49 46 59 ØD 54 2Ø 3D 2Ø 54 42 55 47 ØD
71FØ  ØD 48 49 54 2Ø 5Ø 52 4F 5Ø 45 52 2Ø 4B 45 59 2Ø

72ØØ  54 4F 2Ø 43 4F 4E 54 49 4E 55 45 ØD ØØ ØD 53 45
721Ø  54 2Ø 53 57 49 54 43 48 2Ø 54 4F 2Ø 5Ø 47 4D ØD
722Ø  48 49 54 2Ø 45 4E 54 45 52 2Ø 54 4F 2Ø 43 4F 4E
723Ø  54 49 4E 55 45 ØØ ØD ØD 45 4E 44 2Ø 4F 46 2Ø 43
724Ø  4F 4D 5Ø 41 52 49 53 4F 4E ØD 48 49 54 2Ø 45 4E
725Ø  54 45 52 2Ø 54 4F 2Ø 43 4F 4E 54 49 4E 55 45 ØØ
726Ø  5Ø 52 4F 47 52 41 4D 4D 49 4E 47 2Ø 49 4E 2Ø 5Ø
727Ø  52 4F 47 52 45 53 53 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

Program Listing 1. A hex dump of the 2708 programmer software. Ending address of the program is 7277, and execution address is 7000.

Now LOAD the EPROM contents back into the computer memory and VERIFY that the chip has been properly programmed. If all is well, congratulations are in order. If an error address appears during the VERIFY operation, use your monitor program to determine the difference in the two data blocks starting at 6000 and 6400.

If a data bit in the 2708 is a zero instead of a one, you have no choice but to erase the chip and start over. If a location contains a one that can't be rewritten to a zero, that portion of the chip matrix is probably defective.

Erasure of the 2708 is done by exposing the memory cell matrix under the quartz window on the chip to a strong ultraviolet source. An EPROM eraser, consisting of an ultraviolet lamp and safety interlocks, is available from several companies. The eraser will do a complete job in 20 to 30 minutes.

Programs

Now that you can program 2708s for yourself, what type of program should you place in the EPROM? Remember that the computer can read from the 2708, but cannot write to it with any success.

The program(s) therefore cannot be self-modifying in any form, and do not store temporary data.

There are two good approaches to using programs burned into the 2708. One is to use the program in its present location, with designated locations in dynamic memory used for temporary storage.

Another way of shaping programs entered into EPROM is this: burn a frequently-used program such as a machine-language monitor into the 2708, along with a block-move routine for that program. When you have need of it, enter SYSTEM and the decimal address of the entry point for the block move. Presto! Your monitor program is loaded into dynamic memory and running before you get the cassette version into the recorder.

If you don't have the expansion interface, or equivalent, in your system, you can use addresses in the 3000-37FF range for memory-mapping the programs in EPROM. In fact, you have room for two 2708's in that area. If you do have a disk system, you can still use addresses 3000-33FF for one EPROM.

An alternate address location block for programs in EPROM would be anywhere above the top of your current dynamic memory. For instance, any address block above 7FFF would not interfere with dynamic memory in a 16k system. Since you don't know when additional memory might be included in your system, you might as well locate the EPROM addresses well out of the way to start with (maybe in the Fxxx range).

Another approach to multiple EPROM usage would be the configuring of all address blocks in those EPROM programs to the same range of addresses. Then you could operate several 2708s in parallel, so to speak, selecting the active chip by means of a switch to apply the enabling signals to the desired EPROM.

If you have followed all this so far, you should have no difficulty in constructing a circuit that will accommodate one or more 2708s. Keep in mind that complete address decoding should be provided for, and that the $\overline{RD}$ signal from the TRS-80 should be used to enable the selected 2708 for a read operation.

Building the circuit on another breadboard will be to your advantage, since you already made the cable for the interface board. Unplug that board, insert the 2708 card, and enjoy some freedom from drudgery (encountered in repeatedly loading that often-used program).

Most computer tinkerers can't resist analyzing someone else's experiments and modifying them. That's fine. We'll never get anywhere if we accept ideas without challenging them. ■

THE AFFORDABLE HOME COMPUTER

IS NOW ON SALE.

When PMC-80 was first introduced to the United States, the response was overwhelming! The Computer World was ASTONISHED at the QUALITY, as well as the PRICE. In fact, the PMC-80 has almost all the features of America's best selling computer, the TRS-80, but with a price tag of $200.00 less! (SIMUTEK'S price is $275.00 less!)

Microsoft's Level II Basic and 16K Memory.

Another reason for all the commotion is that the PMC-80 uses the same, easy to learn, LEVEL II BASIC language that the TRS-80 uses! What does this mean? It means that the PMC-80 can run all the 1000's of programs that have been written for the TRS-80 Level II, 16K computer! Some of the programs available include: Flight simulation, World Champion Chess program, Scores of educational and business programs. Word processing programs and hundreds of other games and simulations.

The PMC-80 is expandable!

Your PMC-80 is ready to grow with your needs. Using a special cable, available from Simutek for $35.00, it may be connected to Radio Shack's Expansion interface, to give you up to 48,000 characters of memory, up to 4 disk drives, addition of a telephone communication system, Voice Synthesizer, various printers, a real time clock, as well as plotters and other neat interfaces! As your skills with the PMC-80 improve, you're sure to want some of the ADD-ON's described above. (And these are just a few!)

Save Money! Use your own television!

The PMC-80 has a built in RF MODULATOR so you can use your black and white or color TV for a VIDEO MONITOR! A simple hook-up to your television's antenna connector, makes channel 3 your computer's video channel.

Special Introductory Offer: 25 Free Programs

SIMUTEK, a leading innovator in Home Computer Software, is making a SPECTACULAR INTRODUCTORY OFFER to people that ORDER the PMC-80 NOW. With each purchase, we will give 25 FREE HOME COMPUTER PROGRAMS! Some of these include: Home Amortization tables program, Loan payment programs, Depreciation rate program, Interest table program, Annuity and Investment calculation programs as well as these great animated games: GRAPHIC-TREK 2000: Command the Enterprise!, INVASION WORG.. Stop the invading marauders from space before they take over earth! You command Earth's forces of androids, space fighters, laser guns etc., against the enemy's robots, saucers, proton destroyers, etc!, STAR WARS: Fly your space fighter into the Death Star to destroy it! But watch out, Darth Vader doesn't like you!, SPACE TARGET: A fantastic animated arcade game of skill and daring!, SAUCERS: Can you win win the coveted Medal of Honor?

Comparison Chart

Features	PMC-80	TRS-80
Microsoft's Fantastic Level II Basic	Yes	Yes
Full 128 x 48 Graphics	Yes	Yes
16,000 characters memory	Yes	Yes
Tape recorder for storing or retrieving programs	Yes	Yes
Use your own TV (Save $$)	Yes	No
Expandable to 48,000 characters of in computer memory	Yes	Yes
Use TRS-80 expansion interface	Yes	Yes
Expandable to 4 floppy disk drives (over 100,000 characters of storage on each one!)	Yes	Yes
Telephone Communications available: connect to large computers/electronic mail etc.	Yes	Yes
1000's of ready made programs availble for "educational" and "scientific" applications?	Yes	Yes
Printers available	Yes	Yes
High Speed Z80 CPU	Yes	Yes
Interface available for controlling lights and appliances in home	Yes	Yes
Retail Price	**$645.00**	**$849.00**

Here's what you get:

The PMC-80 microcomputer with 16,000 characters of "In Computer Memory", Microsoft's Level II Basic (built into the computer), a cassette player for storing or retrieving programs or data (cassette player is built into the computer!), an RF Modulator for connecting the PMC-80 to your television set, 25 FREE programs so you start using your computer immediately, complete instruction manual, learning manual and owners manual so you can begin writing your own programs right away!

Best of all, you have the chance to use the PMC-80 in your own home before making your final commitment! Keep it for two weeks, if, for any reason you decide not to become a PMC-80 owner simply send it back, (in new condition please), and we will promptly refund the full amount, including your delivery charge!

Time is of the essence. Please order now, as this price can only be guaranteed through December 25, 1980.

Order Now Save $76.00

Credit card holders may use our TOLL FREE NUMBER. Or send check for $569.00 plus $6.00 delivery. (Arizona residents add $23.80 state tax). Please mention this magazine. (No tax on out of state orders).

Call Toll Free
800-528-1149
In Arizona call 602-886-5880 Collect

SIMUTEK ✓19
COMPUTER PRODUCTS™
9881 E. Skyview Drive
Tucson, AZ 85710

TRS-80 is a registered trademark of Radio Shack, a Tandy Corp.

✓ Reader Service—see page 242

Riding the right freight to the stars?
Then you'll be interested in this review of astronomical software.

Starstuff

Russell M. Genet
Fairborn Observatory
1247 Folk Rd.
Fairborn, OH 45324

Twinkle, Twinkle Little Star
My '80 Knows Right
Where You Are. . .

Based on the overwhelming response I received to the short "80 Skywatch" note in the May 1980 issue of *80 Microcomputing*, there must be a sizable segment of 80 owners who desert their faithful computers on clear nights to peer through telescopes at stars, planets, and satellites. As adult toys, telescopes compare favorably with microcomputers in both expense and fascination, and one can easily appreciate the dilemma of trying to decide which toy to play with. This can easily lead to depression and eventually, severe psychosis. Fortunately, American ingenuity has risen to the challenge and the 80 skywatcher need no longer dislocate his psyche with wrenching indecision, but can play with his two favorite toys simultaneously.

The first program comes from George Hall, of Asheville, NC. For less than $20, well under the cost of a single psychiatric session, you can load George's cassette in your 80 and convert local to sidereal time so you can adjust your setting circle. If you're not sure when the sun will set, due to a horizon obscured with skyscrapers, George's program and your trusty 80 will keep you posted, throw in the time of twilight, moonrise, and moonset, and also give you the moonphase. The position of the sun, moon, and planets are all a snap, with numerical positions and a sky map display. If you see a strange object, the UFO identifier will quickly tell you whether it is in earth orbit and thus a satellite or something really weird. If you consistently get OBJECT NOT IN EARTH ORBIT, either send the program back to George or start seeing your psychiatrist again!

The second program comes from SAT TRAK International, Colorado Springs, CO. This program is so professional, it is rumored that the government's big satellite computing center in nearby Cheyenne Mountain isn't really run by two back-to-back CDC Cybers, but is actually taken care of by an 80 tucked away in the janitor's closet churning away on a SAT TRAK program. With this program, your 80 can chart the positions of as many as 50 different satellites at any time in the past, present, or future.

The latitude, longitude, altitude, right-ascension, declination, range, azimuth, and elevation of satellites are all available. On cloudy nights, you can display the satellite tracks on a world map. Ephemerides (also known as orbits) are easily updated from data provided free by NASA. The amateur observer who claimed he saw a Russian through a spaceship porthole with his Celestron-8 needs to head back for an extra couch session, however.

Program number 3 is provided by Cosmic Computer Works, Belmont, MA. The works you can get includes an almanac program that will provide you with all you want to know on the whereabouts of the sun and moon, times of sun and moon rise and set, twilight, etc. It covers AD 1700 to 2200, thus being useful to your distant descendants, and is guaranteed to work in Antarctica if your fingers don't freeze to the keyboard. For the comet chaser, Ephem will keep you in touch with your favorite ball of gas and rock, and will even tell you at what earthly latitude the comet will be highest in the sky so you can properly plan your vacation. The 80 equipped comet-chaser should avoid Upper Slabovia, however, as they tend to impound computers as secret imperialist weapons. The final word from the Works is Myoptics. This program is just the thing for those amateurs who not only grind their own telescopic optics, but wish to design them also. With Myoptics, your 80 will design paraboloidal, ellipsoidal, or hyperboloidal mirrors or lenses, and Schmidt-Cassegrains and Eccentric-pupil Herschelians will be a snap. Be forewarned,

Super Software! Super Introductory Savings!

THE WIZARD	ORCHESTRA 80	SUPER-HOST
TRS-80 Model I 32K **DISK**	**A Music Synthesis System** **by Jon Bokelman** **TRS-80 16K Level 2**	**TRS-80 MODEL I 32 K** **DISK**

LATEST FROM AUTHOR OF TRS-OPERA, RICHARD TAYLOR!

Challenge your companions to a fast-paced, question-and-answer game coupling sound effects and non-stop graphics to test your expertise on five subjects. THE WIZARD'S a wiz at assimilating new subjects . . . its built-in utility program lets you create even more subjects to give THE WIZARD a mind of your own. Timed responses give each player a chance to steal from the opponents or force other players into answering. Match wits with THE WIZARD, a Programs Unlimited exclusive!

Regularly $19.95

YOU HAVE TO HEAR IT TO BELIEVE IT!

Turn your 16K TRS-80 Level II into a high quality musical instrument. The software includes a digital synthesizer, a music language compiler, a full screen editor, a file manager and a voice initialization utility. The hardware is a single 1½ x 2" PC board which plugs into the expansion connector on the TRS-80 Keyboard or the screen printer connector on the expansion interface. Plugs into any amplifier. Write your own music or join scores of others in trading music.

Regularly $79.95
FREE DISK OF MUSIC!

BREAKTHROUGH! UNLIMITED ACCESS TO YOUR TRS-80!

The finest HOST program on the market and a Programs Unlimited **exclusive.** • Runs on TRSDOS, NEWDOS, NEWDOS-80 • Data-in and Data-out translation tables • Correctly advances the date on all DOS's • Masks out undesirable characters (i.e. BREAK) • Remembers the date and time after Reboot • Lowercase driver option • Optional Password Entry • Automatic disconnect upon loss of carrier • Sets Nulls and Line Feeds • Plus, many more outstanding features!

Regularly $29.95

Meet PROGRAMS UNLIMITED, the microcomputer industry's newest and most exciting source for the hobbyist who demands the widest possible choice of top quality programs!

★*MENTION "80 MICROCOMPUTING" WHEN YOU ORDER, AND SAVE $5 ON OUR MOST POPULAR PROGRAMS.*

HOW TO ORDER

OUTSIDE NY STATE – PHONE TOLL FREE (800) 645–6038 (9 AM - 5 PM)
NY STATE ONLY – PHONE COLLECT (516) 997-8668 (9 AM - 5 PM)
COMPUTER – PHONE (516) 334-3134
MAIL – DEPT. M, P.O. BOX 265, JERICHO, NY 11753

MasterCard and Visa accepted. Add $1 for postage and handling.
No sales tax for out-of-state residents.

★ **ASK FOR YOUR FREE COPY OF THE PROGRAMS UNLIMITED CATALOG.**

the software source™
PROGRAMS UNLIMITED™

TRS-80 and TRSDOS are registered trademarks of TANDY CORPORATION
NEWDOS-80 and NEWDOS 2.1 are registered trademarks of Apparat.

✓110

All orders are guaranteed. Watch for PROGRAMS UNLIMITED RETAIL CENTERS opening soon from coast to coast.

✓Reader Service—see page 242

however, that many an amateur has gone over the deep end trying to get a perfect hyperboloidal figure. They don't call them telescope nuts for nothing.

The final cure is Uranographia Machina, from Luninscript Inc., Rochester, NY. This program contains detailed data on 1400 stellar objects. GC and Bayer identifications are given, along with position, magnitude, distance, and spectral class. The position of the sun, moon and planets against the star background can be displayed. An 80 disk version is in the works. Observing 1400 stellar objects should provide a lengthy cure indeed.

Several cases of successful self-therapy have been reported. Among these were Sidney Freidin, Laredo, TX, who wrote a program for his 80 to give the altitude and azimuth of the 25 brightest stars, to aid in aiming his Celestron C-8. Another is Bob Patterson, North Little Rock, AR. Bob uses his 80 to position his radio telescope. Dr. E. P. Belserene, a professional astronomer at Maria Mitchell Observatory, uses her 80 to catalog the observatory's extensive collection of photographic plates, and to calculate heliocentric time.

A former schizoid myself on clear nights, I connected my 80 through a six-chip-interface-real time clock to a photoelectric photometer on my 8-inch Cassegrain. I observe stars with small but interesting light variations with it, which I report to professional astronomers. My 80 tells me which stars to point at, and where to place the photometer controls (very handy at 4 AM). It also displays a trace of the light from the star as it is received, twinkles and all, automatically records the light intensity and time, and computes the stellar magnitudes. My 80 is also used to transform the magnitudes to standard form, add heliocentric corrections, do a least-square-solution for time of minimum light in eclipsing binary stars, and predict the times of eclipse in double precision.

Unlike the rare solar eclipses, some binary stars have an eclipse every six hours. If this sort of thing turns you on, your recovery will be hastened by joining the International Amateur-Professional Photoelectric Photometry Communication (IAPPPC), c/o Dr. D. S. Hall, Dept. of Astronomy, Vanderbilt Univ., Nashville, TN 37235, and the American Association of Variable Star Observers (AAVSO), c/o Janet Mattei, 187 Concord Ave., Cambridge, MA 02138.

I have heard unconfirmed rumors of a former sufferer of acute Psychotic Astronomitus-Computeritus that programmed his 80 to run his telescope in a completely automatic mode. Now every clear night he happily plays blocks with his little boy, walks the dog, and has restored his marriage to its pre-80 bliss. ■

EDAS EDAS

A sophisticated Editor & Assembler setting the standard for the '80 Model I & Model III. All EDAS commands and SOURCE text can be entered in either upper case or lower case. Direct assembly form memory or disk by means of *GET assembler directives. This gives text buffer capacity equal to your drive configuration! 30,000 bytes of symbol table.

Direct assembly to disk or memory for faster debugging operations! DOS "system" command functions KILL, DIR, FREE, and LIST are available from within the environment of EDAS.

The Editor, with renumber, maintains command syntax identical to the BASIC editor. Global change permits you to alter a string throughout a designated range of lines while block move relocates lines of text.

EDAS is priced at $79 plus $3 S&H. A 72-page manual included.

cmdfil

Now you can append two or more CMD files and/or SYSTEM tapes. Perform transfer to & from disk/tape of SYSTEM/CMD modules with offset capabilities. Read VTOS ISAM overlays. More! $20

MISOSYS SERIOUS SOFTWARE

LDOS is the latest generation of operating systems for the TRS-80 computers. LDOS is completely documented in an extensive operating manual and total on-going support from some of the most knowledgeable people in the industry is provided. In short, LDOS is THE operating system to use with the TRS-80 computers!

$139.00
+$3 S&H

LDOS 4.1

dsmblr

Complement your assembly language tools with this Z80 disassembler which produces screen, printer, cassette, or disk file output. A two-pass process provides SYMBOLS for 16-bit address and 8-bit relative references. EQUates & ORG are generated. Read SYSTEM programs & display load address range. $20 (DSMBLR I for non-disk use is $15)

Specify Model I or Model III

MISOSYS - Dept MU
5904 Edgehill Drive
Alexandria, Virginia 22303
703-960-2998 MicroNET 70140,310
Dealer Inquiries Invited ✓221

diskmod

Turn your Editor Assembler into a disk package. This 32K patch modifies EDTASM for DOS operation. Features? Add full disk I/O, block move, global change, printer pagination with optional prompting, sorted symbol table, print memory utilization, correct DEFM expansion, protect memory, and recover after BOOT. From within the EDTASM you will have DIR, KILL, & FREE. Upgrade your EDTASM today! Version for EDTASM+ coming soon. $20.

THE BØØK

THE BØØK must be a part of your Z-80 language tools. Volume I gives you access to all math operations in your Level II ROM including ASCII-Binary conversions. Included is a symbol table of the entire machine noting over 500 addresses. Volume II tells you everything you wanted to know about the Level II I/O - printer, keyboard, video, and cassette routines are fully explained. Each volume has a fully-commented listing of all the routines discussed. THE BØØKs will save you hours of assembler program development time. Don't start programming without THE BØØKs. Each volume is priced at $14.95 + $1.50 S&H or buy both for $24.95.

NOW AVAILABLE FROM THE ALTERNATE SOURCE

Percom Doubler	$200.00
Newdos/80 Double-Zap II	45.00
SAFOR	70.00
Olympic Decathlon	22.50
VTOS 4.0	90.00
NEWDOS/80	135.00
EDAS	70.00
Word Slinger	25.00
Disc Interfacing Guide	4.50

Cash Sales Only/Shipping Included!!!

DON'T FORGET a subscription to THE ALTERNATE SOURCE

The rapidly evolving technical newsletter for the TRS-80
$12.00 per year U.S. only

BTI

off the cuff information about the TRS-80
$7.00 per year U.S. only
each issue mailed first class

THE ALTERNATE SOURCE
1806 Ada Street ✓138
Lansing, Michigan 48910
Ph. 517/487-3358 or 485-0344

SPECIALSPECIAL**

TRS-80 ADD ON DRIVES IMMEDIATE DELIVERY

SINGLE SIDED $225.00
DOUBLE SIDED $345.00

COMPLETE SYSTEMS
SINGLE SIDED $365.00
DOUBLE SIDED $485.00

INCLUDES:
- MINI DISK DRIVE
- FUSED POWER SUPPLY
- VENTED CABINET
- CABLE
- 90 DAY WARRANTY
- FACTORY ASSEMBLED
- FACTORY TESTED

THESE ARE NEW 5" FD's

I 2 INTERFACE, INC ✓246
20932 CANTARA ST
CANOGA PARK, CA 91304
(213) 341-7914

VISA AND MASTER CHARGE ACCEPTED

Let Your TRS-80® Teach You ASSEMBLY LANGUAGE

Tired of buying book after book on assembly language programming and still not knowing your **POP** from your **PUSH**?

REMSOFT proudly announces a more efficient way, using your own TRS-80®, to learn the fundamentals of assembly language programming --at YOUR pace and at YOUR convenience.

Our unique package, "**INTRODUCTION TO TRS-80® ASSEMBLY PROGRAMMING**", will provide you with the following:

- Ten 45-minute lessons on audio cassettes.
- A driver program to make your TRS-80® video monitor serve as a blackboard for the instructor.
- A display program for each lesson to provide illustration and reinforcement for what you are hearing.
- A textbook on TRS-80® Assembly Language Programming.
- Step-by-step dissection of complete and useful routines to test memory and to gain direct control over the keyboard, video monitor, and printer.
- How to access and use powerful routines in your Level II ROM.

This course was developed and recorded by Joseph E. Willis and is based on the successful series of courses he has taught at Meta Technologies Corporation, the Radio Shack Computer Center, and other locations in Northern Ohio. The minimum system required is a Level II, 16K RAM.

REMASSEM-1 only $69.95

REMSOFT, Inc.
571 E. 185 st.
Euclid, Ohio 44119
(216)531-1338

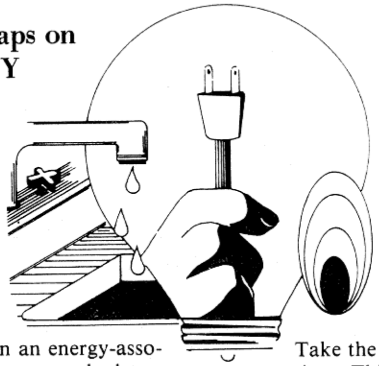
Include $1.50 for shipping and handling. ✓70
Ohio residents add 5½% sales tax.
TRS-80® is a trademark of the Tandy Corp.

✓Reader Service—see page 242

These Penny-Pinching Programs Help You Beat High Energy Costs!

Designed for Home and Commercial Use

Here's your chance to put the wraps on staggering energy costs. ENERGY AUDIT helps you inspect any home, analyze your findings and make fuel-conservation investments that result in big savings.



ENERGY CONSUMPTION lets you monitor and manage fuel use for maximum cost efficiency. This combination will mean more dollars in your pocket in these uncertain times.

1. Energy Audit

Whether you're a home owner or involved in an energy-associated business, this is a "must" program for accurately determining heat loss. This program creates a computerized model of any dwelling. It will describe what materials are needed and the estimated construction costs to make a building an energy miser. Requirements: TRS-80, Level II, 16K; Expansion interface with at least 16K; 1 mini-disk drive; and a compatible DOS for disk-based version; and an optional printer. **Order No. 0052RD, disk, $75.00 or 0089R, cassette, $49.95.**

2. Energy Consumption

Take the guesswork out of energy consumption and conservation. This program keeps utility bills for a 5-year period. Records precise amounts used and costs for natural gas, water and electricity. Keep track of energy costs, examine seasonal fluctuations and evaluate conservation efforts you've undertaken. The soaring cost of fuel requires careful energy management. This program can make you a tight-fisted professional! TRS-80, Level II, 16K. **Order No. 0132R $9.95.**

Instant Software™

PETERBOROUGH, N.H. 03458 ✓2

THEY'RE EASY TO ORDER. . .
- See Your Instant Software Dealer, or
- Call Toll-Free 1-800-258-5473.

SAVE YOUR TIME ...AND YOUR MONEY.

Instant Software has the two best mail list programs available for your TRS-80 Model I and Model II.

Mail/List for Model I and Model II

This mailing list program maintains separate alphabetical and zip code files in *constant sort.* When you add a name to your list, it will *automatically* be inserted into its correct position in the files; therefore it's always ready to print labels!

It will record your information in these fields: NAME, ADDRESS, CITY, STATE, ZIP, PHONE NUMBER, PHONE EXTENSION and a five character CODE field. You have the choice of a 3 line, 4 line, or user defined label format. It can even include (optionally) a message line on your label!

The programs most outstanding feature is its sorting capabilities. Mail/List allows you to choose which names you want to be printed from the whole list. For example, all people in one zip code, or all people named Jones, who are living in a particular city or state. For any name in your list you can assign a code within the CODE field. You can then specify the code when printing labels, and only names with that code will be printed out. You can specify up to 9 different codes!

Every business and organization will save time and money with Mail/list to keep track of customers or members.

TRS-80 Model I version, Order No. 5000RD $99.00.
Requires 16K RAM, Expansion Interface with at least 16K RAM, one disk-drive and a printer.

TRS-80 Model II version, Order No.5001RD $199.00.
Requires 64K RAM and printer.

WRITE FOR OUR NEW CATALOG

TO ORDER: See Your Local Instant Software Dealer or Call Toll-Free 1-800-258-5473.

One-D Mailing List

Here is a mailing list system that can be run on only ONE disk-drive! You can have up to 17 fields of selection for name/address retrieval.

•Disk versatility allows you to add, delete, or change the numerous details stored in the system.

•Features of the One-D Mailing List includes:

•Automatic name sort (alphabetically or by ZIP code). •Rapid access to any name on file. •Easy error correction and recovery. •Prints selective name listings. •Revise or update listings at any time. •Up to 2500 names on-line (with 4 drives). •Prints a list of all names on file. •Prints mailing labels.

This package requires the following minimum system: 1. A TRS-80 Model I Level II microcomputer with 16K RAM. 2. An Expansion interface with 0 to 32K of RAM. 3. A single disk drive (extra drives optional). 4. A printer. 5. Any TRSDOS compatible Disk Operating System.

Order No. 0123RD $24.95. ✓2

Instant Software™

PETERBOROUGH, N.H. 03458

René Descates * Homer * Jeremy Bentham * Plato * John Adams * Pliny the Elder *

Galileo Galilei * Sigmund Freud * Michelangelo * Ludwig Van Beethoven * Leibtniz * Thomas Hobbes *

Computer\n

1: Device designed to execute a sequence of mathematical operations.

Education for the Home

Beginner's Russian

This package consists of three programs that graphically display the Cyrillic alphabet. The programs are arranged so that you progress from one to the next—building your knowledge as *you* progress. It includes instructions on proper pronunciation of the letters and even an introduction to simple Russian words.
Order No. 0136R $9.95

Everyday Russian

Everyday Russian will acquaint you with the Russian words relating to: foods, places to eat, everyday signs, and the names of common stores. You will also learn the order of the Cyrillic alphabet. Each of the three divisions of this package will teach you the words and then quiz you on comprehension. You can even practice typing in Russian, using your TRS-80 keyboard as a "Cyrillic typewriter."
Order No. 0137R $9.95

The Russian Disk

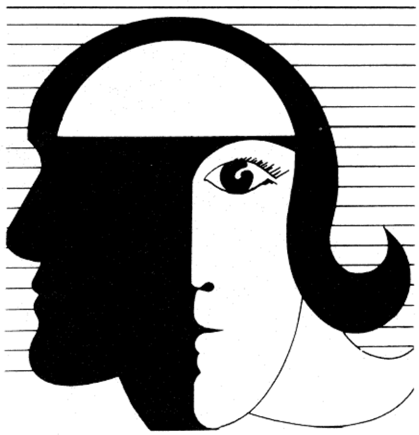
Now you can have *both* the Beginner's Russian and Everyday Russian packages on floppy disk! Requires an Expansion Interface with 16K and one disk drive.
Order No. 0212RD $24.95

Teacher

This program allows you to input any number of questions and answers. The computer will prepare tests, give quizes, provide up to three hints per question and even give (optional) graphic rewards for correct answers. Perfect for parents, teachers, or anyone faced with learning a lot of data in a short time.
Order No. 0065R $9.95

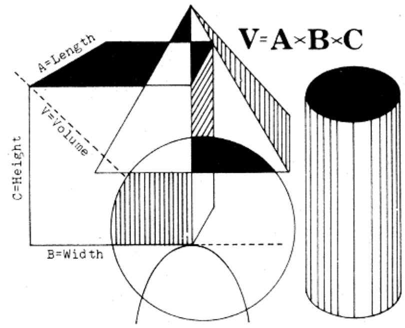
Wordwatch

Four programs for budding lexicographers, etymologists, or anyone else who uses words. In WORD RACE, you must choose the proper definitions. Find the misspelled word in HIDE N SPELL. Take a pre-recorded quiz in SPELLING BEE, in which the words are played aloud! Meet variations on proper spelling in SPELLING TUTOR.
Order No. 0111R $7.95



IQ Test

Are you smart enough to buy this package? IQ Test will administer and score an intelligence test in 30 minutes flat! There are three equivalent tests, each consisting of 35 questions, designed to test your general knowledge and problem solving abilities. Most of us claim a "touch of genius"—here's your chance to prove it!
Order No. 0157R $9.95



Archimedes' Apprentice

A tutorial software package that will teach you the formulas used to find the volume of *any* solid object. It covers parallelopipeds (cubes and rectangular solids), prisms, pyramids, cylinders, cones and spheres. It can even quiz you on how well you learned the lesson.
Order No. 0092R $9.95

Video Speed-Reading Trainer

You can increase your reading speed and comprehension. How? By practicing, that's how! This three-part program will flash characters or words on the screen, then you must echo what you saw. You can begin at a relatively slow rate, because the computer will advance your speed automatically as your speed and comprehension increase. It will train you with numbers, letters, words and phrases.
Order No. 0100R $9.95

Typing Teacher

A complete seven-part package that guides you from familiarization with the keyboard, through typing words (and phrases), to mastery of touch typing. Your video monitor becomes a bottomless page for typing practice!
Order No. 0099R $9.95

All packages listed are for the TRS-80 Model I Level II; they require 16K of memory and are cassette-based unless otherwise indicated.

Instant Software™ PETERBOROUGH, N.H. 03458
603-924-7296

Rebbe Akiva * John Stuart Mill * Pythagoras * Niccolo Machiavelli * Sun Tzu Wu *

An elemental application Avogadro would envy.

Molality Crunching

Dr. Robert Suder
Portage Northern High School
1000 Idaho
Portage, MI 49009

We will see how atomic masses can be used in a practical problem on the computer, since many chemistry calculations use an atom's mass and the TRS-80 would be helpful. I've placed values in the computer as a subroutine.

Background

For some two thousand years, most people believed the teachings of Aristotle concerning the nature of matter, that matter is continuous. That is, it can be divided into increasingly smaller units and still retain its original properties.

Then in 1803, an English schoolteacher named John Dalton concluded that all evidence indicates that Aristotle is incorrect. There is a particle of matter that cannot be divided further, and referred to this particle as the atom from the Greek word *atomos*, meaning indivisible.

Dalton also said these atoms would have different masses depending on the kind of element they came from.

In 1811, the Italian physicist Amadeo Avogadro suggested a method to determine the relative mass of an atom. We use relative mass because it is impossible to measure the mass of an atom directly.

Scientists selected the carbon atom as an arbitrary standard and compared the masses of all other atoms to it. It was decided to assign a value of 12 as its relative mass. Using the relative mass scale developed by Avogadro and others, chemists were able to determine the relative masses of molecules. For example, if sodium (Na) and bromine (Br) have relative masses of 23 and 80, respectively, the relative mass of the sodium bromide (NaBr) molecule is $23 + 80$ or 103. The term relative mass is now called atomic mass for atoms and molecular mass for molecules.

Atomic Program

This program will compute the percent composition of a molecule, or just what part of the total mass is contributed by each atom. To do this, we need our atomic mass table which begins at line 10000.

Let's find the percent composition of magnesium bromide ($MgBr_2$). The magnesium and bromine atoms have atomic masses of 24 and 80, respectively. Therefore, the molecular mass of $MgBr_2$ is $24 + 2(80)$ or 184. The percent Mg is $(24/184) \times 100$ or 13%, and the percent Br is $((2)(80)/184) \times 100$ or 87%. The printout is shown in Fig. 1.

```
THE MOLAR MASS IS 184.113 G/MOL
THE PERCENT MG IS 13.2011
THE PERCENT BR IS 86.7989
```

Fig. 1

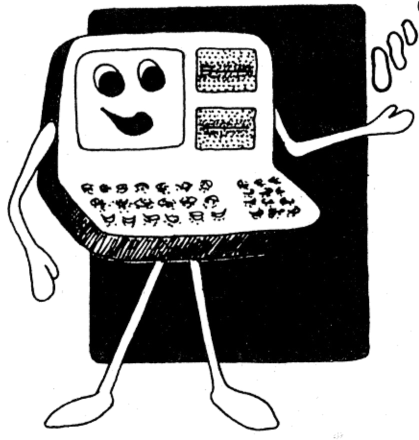
```
THE MOLAR MASS IS 303.326 G/MOL
THE PERCENT AU IS 64.9357
THE PERCENT CL IS 35.0643
```

Fig. 2

Another example: Assume that you have 2.0 kilograms of the compound gold (III) chloride ($AuCl_3$) and want to know how much gold it contains. The printout in Fig. 2 tells us that it contains 65% gold. Sixty-five percent of 2.0 kg is $2.0 \times 0.65 = 1.3$ kg of gold being in the sample.

The atomic mass table is a subroutine since it is used often. The computer scans the list of elements beginning at line 10000. It is a good idea to place the more common elements first. This way, less time will be needed to find the elements used most. Naturally, one can continue the table with as many elements as desired. I have just included a few.

The program to determine the percent composition is clear. First, the symbol of the element is read. This value is then multiplied by the number of atoms present. Then the process is repeated for the remainder of the atoms. These values are summed, thus giving the molecular mass of the compound. The actual computation of the percent composition is done in the print statement.■



LOOK MA!! I HAVE DRIVES!!

TRS-80* Model III now available
with full range of M.P.I.
Double Density Drives
up to 2.8 Meg'bytes on-line
floppy disk storage.

Also available - Ultra DOS

CALL FOR OUR UNBELIEVABLE LOW PRICING

LEVEL IV PRODUCTS, INC. ✓14

32238 Schoolcraft Road, Suite F4 • Livonia, MI 48154
313-525-6200 Outside Michigan call 1-800-521-3305

Dealers Orders Welcome

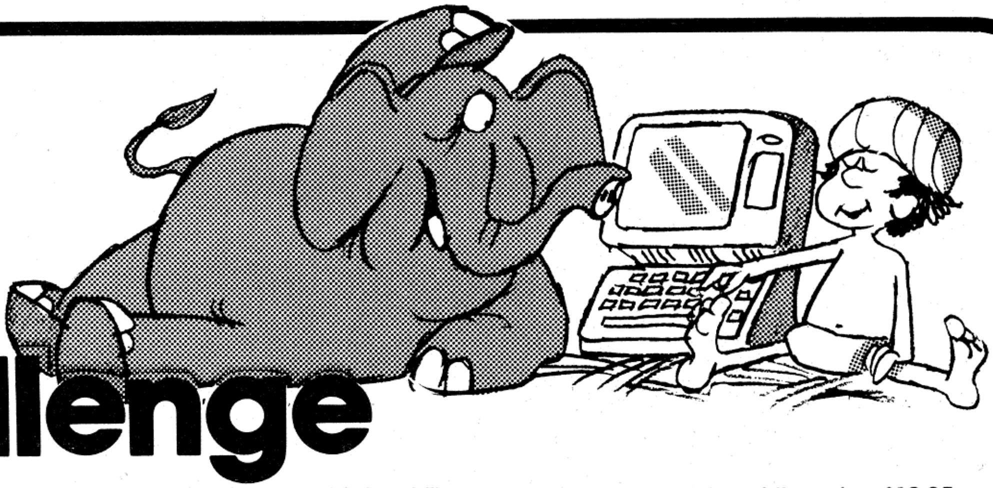
Please add $2.50 for shipping and handling.
$1.50 C.O.D.

**Product of the Tandy Corporation*

Level IV Products Catalog
NEW - SEND $2 FOR YOUR COPY
REFUNDABLE ON FIRST ORDER

master charge
VISA

the kim challenge



From Rudyard Kipling's KIM, General Computer brings you an adaptation of the exciting, mind expanding game of memory and recall. KIM uses dynamic handicapping to compensate for skill differences while urging each player into greater challenges. Everything adjusts --display times, number of objects displayed, identification difficulty, and even scoring as you play your way through a data-base of thousands of items. Quicken your perception, sharpen your awareness, and develop an elephant-like memory ... while enjoying the competitive excitement of playing KIM.

- An exciting two player game with competitive skill-difference handicap scoring (Junior could beat daddy every time!)
- Or, A challenging single-player contest
- Includes a BASIC source listing as part of a trend-setting manual

- 16K, Level II version $19.95 (cassette)
- 32K, TRS-DOS two drive version $24.95 (diskette)
- For Visa and Mastercharge orders, call toll free anytime 1-800-824-7888. In California, 1-800-852-7777, ask for Operator 115
- Dealer Inquiries invited

General Computer Co.
4873 Langer Lane
Woodbridge, VA 22193
✓251

MORE FOR YOUR RADIO SHACK TRS-80 MODEL I !

THE DATAHANDLER

DATABASE MANAGEMENT SYSTEM IN MMSFORTH

Now the power, speed and compactness of MMSFORTH drive a major applications program for many of YOUR home, school and business tasks! Imagine a sophisticated database management system with flexibility to create, maintain and print mailing lists with multiple address lines, Canadian or the new 9-digit U.S. ZIP codes, and multiple phone numbers, plus the speed to load hundreds of records or sort them on several fields in 5 seconds! Manage inventories with selection by any character or combination. Balance checkbook records and do CONDITIONAL reporting of expenses or other calculations. File any records and recall selected ones with optional upper/lower case match, in standard or custom formats. Personnel, membership lists, bibliographies, catalogs of record, stamp and coin collections—you name it! ALL INSTANTLY, without wasted bytes, and with cueing from screen so good that non-programmers quickly master its use! With manual, sample data files and custom words for mail list and checkbook use.

Technical: Handles data as compressed indexed sequential subfiles of up to 25K characters (9K in 32K RAM). Access 1-4 data diskettes. Modified Quicksort. Optionally precompiles for 5-second program load. Self-adjusts for many routine mods. Structured and modular MMSFORTH source code ideal for custom modifications.

THE DATAHANDLER V1.1, a very sophisticated database management system operable by non-programmers (requires Disk MMSFORTH, 1 drive & 32K RAM); with manuals **$59.95***

MMSFORTH

THE PROFESSIONAL FORTH FOR TRS-80 MODEL I

(Over 1,000 systems in use)

MMSFORTH Disk System V1.9 (requires 1 disk drive & 16K RAM) . . . . just **$79.95***
MMSFORTH Cassette System V1.8 (requires Level II BASIC & 16K RAM) **$59.95***

AND MMS GIVES IT PROFESSIONAL SUPPORT

Source code provided
MMSFORTH Newsletter
Many demo programs aboard
MMSFORTH User Groups
Programming staff can adapt THE DATAHANDLER to YOUR needs.

MMSFORTH UTILITIES DISKETTE: includes FLOATING POINT MATH (L.2 BASIC ROM routines plus Complex numbers, Rectangular-Polar coordinate conversions, Degrees mode, more), plus a full Forth-style Z80 ASSEMBLER; plus a powerful CROSS-REFERENCER to list Forth words by block and line. All on one diskette (requires MMSFORTH, 1 drive & 16K RAM), . . **$39.95***

FORTH BOOKS AVAILABLE

MICROFORTH PRIMER (comes with MMSFORTH) separately **$15.00***
USING FORTH — more detailed and advanced than above **$25.00***
THREADED INTERPRETIVE LANGUAGES—advanced, excellent analysis of MMSFORTH-like language **$18.95***
CALTECH FORTH MANUAL — good on Forth internal structure, etc **$10.00***

* — Software prices include manuals and require signing of a single-system user license. Add $2.00 S/H plus $1.00 per additional book; Mass. orders add 5% tax. Foreign orders add 15%. UPS COD, VISA & M/C accepted; no unpaid purchase orders, please.

Send SASE for free MMSFORTH information. Good dealers sought.

Get MMSFORTH products from your computer dealer or

MILLER MICROCOMPUTER SERVICES (M2) ✓112
61 Lake Shore Road, Natick, MA 01760
(617) 653-6136

Introducing

COBOL + FORTRAN + 64K RAM

FOR YOUR TRS-80*

★Release your software chains with the *NEW* ***FREEDOM OPTION***, a plugable change that restructures the TRS-80* on command to perform like a large Z80 system. All the TRS-80* features are retained. All TRS-80* software will run without interference. The option is supplied with a fully assembled & tested ***FREEDOM BOARD, T8/OS*** on a 5¼" disk, and complete instructions. ***T8/OS*** allows your TRS-80* to execute most software originally written for ***CDOS***[1]***, TSA/OS***[2]***,*** and ***CP/M***[3] operating systems. ***T8/OS*** opens the door to higher level languages and existing programs.

★To further enhance your TRS-80* processing power, a ***MEMORY EXPANSION OPTION*** is available to replace, on command, the *ROM* and provide *A FULL 64K RAM*. This option is switched into operation by the ***FREEDOM BOARD*** providing *57K* of *USER RAM* with ***T8/OS*** loaded. Both options are fully assembled & tested and fit into the TRS-80* keyboard enclosure. Write for more details. 6 Mo. Board Warranty.

FREEDOM OPTION. $245
MEMORY EXPANSION OPTION. . . . $295

Send Check or Money Order to:
(MASS. RESIDENTS, PLEASE INCLUDE 5% TAX)

F.E.C. ltd. ✓141

P.O. Box 2368 • Woburn, MA. 01888
(617) 944-5329

VISA   master charge

*TRS-80 ® Tandy Corp
1-CDOS ® Cromemco Inc.
2-TSA/OS ® TEA Software Inc.
3-CP/M ® Digital Research Inc.

◂ ************ **NEW** ************ ▸

◂ THE PROGRAM YOU'VE BEEN WAITING FOR ▸

A magazine index for ALL your 1980 issues of 80-MICROCOMPUTING or KILOBAUD MICROCOMPUTING!

Locate and use those special tutorials and how-to articles that you never read because you 'weren't interested' at the time. Find all articles on printers, discs ... you name it! FAST & EASY access of any article by subject search OR keyword search, indexed by topic. NO catchy magazine titles to remember!

All programs are fully documented, guaranteed to load. ORDER NOW at a low introductory price and put your magazines to work for **YOU!**

PLEASE SPECIFY MAGAZINE AND DATES!

80-Microcomputing OR Kilobaud.

Jan-June	1980	16K	L-II	$11.95
Jul-Dec	1980	16K	L-II	$11.95
Jan-Dec	1980	32K	L-II	$19.95

Send check or M.O. to:

THE SYSTEMS SHACK
P.O. Box 281 ✓157
Flushing, N.Y. 11358

N.Y.C. Residents add 8% tax
Kilobaud & 80-Microcomputing published by Wayne Green, Inc.
****

WE'RE OVERSTOCKED

SAVE 25%!

IRV
programmable keys/screen editor

Usually $25.00
Until March 1, 1981 . . . . $18.75

SAVE 25%

BXREF
basic cross reference & formatter

Usually $29.95
Until March 1, 1981 . . . . $22.46

SAVE 25%!

Z80ZAP/CMD
a disk modification utility

Usually $29.95
Until March 1, 1981 . . . . $22.46

THE ALTERNATE SOURCE
1806 Ada Street ✓138
Lansing, Michigan 48910
Ph. 517/487-3358 or 485-0344

Add $2.00 for First Class Delivery

CalData Systems Presents ✓294

WordMagic II

Complete WORD PROCESSING designed specifically for
The Radio Shack TRS-80 Model II Computer

WordMagic II™ is a Word Processor designed specifically for the Radio Shack TRS-80 Model II Computer.*

FEATURES INCLUDE:

- Mailing List/Labels Generation
- Automatic Merging of Mailing Data with Text Files to created "PERSONALIZED" Form Letters
- Automatic wrap-around in text entry
- Margination, Paging, Complete Cursor Movement
- Complete Editing Commands—Insertion, Global Substitution, Overwrite, etc.
- Centering, Smooth Right, Left Justify
- Table of Contents Generation
- Automatic Page Numbering
- Variable Form Lengths
- Underlining
- Line Numbering

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

Requires 64K Model II, TRSDOS & BASIC (not provided with WordMagic)
Introductory Pricing: $195.00 Manual $20. (Cal. Res. add tax)

CalData Systems   P.O. Box 178446
San Diego, CA 92117 (714) 272-2661

```
10 REM  PROGRAM FOR DETERMINING PERCENT COMPOSITION OF
    A
20 REM  MOLECULE
30 REM        BY  DR. ROBERT SUDER
40 CLS
50 INPUT "DO YOU WANT INSTRUCTIONS?  TYPE Y (YES) OR N"
    ;S$
60 IF S$ = "N" THEN 200
70 CLS
80 PRINT "USING A DATA STATEMENT, ENTER THE FORMULA OF
    THE"
90 PRINT "COMPOUND, SEPARATING EACH SYMBOL AND NUMBER W
    ITH A"
100 PRINT "COMMA.  FOR EXAMPLE, WATER IS ENTERED AS"
110 PRINT "     200 DATA H,2,O,1"
120 PRINT
130 PRINT " GLUCOSE WOULD BE    200  DATA C,6,H,12,O,6"
140 PRINT
150 PRINT "USE  200 AS THE DATA LINE"
160 PRINT
170 PRINT "AFTER ENTERING THE DATA, TYPE RUN"
180 END
190 PRINT
200 DATA AU,1,CL,3
210 PRINT "IT IS ASSUMED THAT YOU HAVE THE MOLECULAR FO
    RMULA"
220 PRINT "IN THE DATA STATEMENT.  IF NOT, YOU HAD BETT
    ER GET"
230 PRINT "INSTRUCTIONS BEFORE CONTINUING.  IF YOU WANT
    "
240 PRINT "MORE INSTRUCTIONS, TYPE Y.   OTHERWISE TYPE
    N."
250 INPUT S$
260 IF S$ = "Y" THEN 70
270 PRINT
280 INPUT "HOW MANY DIFFERENT ATOMS ARE PRESENT";NN
290 FOR N = 1 TO NN
300 READ A$(N)
310 A$ = A$(N)
320 GOSUB 10000
330 READ A(I)
340 M(N) = M * A(I)
350 MM = MM + M(N)
360 NEXT N
370 CLS
380 PRINT "THE MOLECULAR MASS IS";MM;"G/MOL"
390 PRINT
400 FOR N = 1 TO NN
410 PRINT"THE PERCENT ";A$(N);" IS"((M(N)/MM)*100)
420 NEXT N
430 PRINT
440 PRINT " IF YOU WOULD LIKE TO DETERMINE ANOTHER PERC
    ENT"
450 PRINT "COMPOSITION, ENTER THE FORMULA IN THE DATA S
    TATEMENT"
460 PRINT "ON LINE 200 AND TYPE  RUN"
470 END
10000 IF A$ <> "H" THEN 10030
10010 M = 1.0079
10020 RETURN
10030 IF A$<>"C" THEN 10060
10040 M = 12.011
10050 RETURN
10060 IF A$<>"O" THEN 10090
10070 M = 15.9994
10080 RETURN
10090 IF A$ <> "N" THEN 10120
10100 M = 14.0067
10110 RETURN
10120 IF A$ <> "CL" THEN 10150
10130 M = 35.453
10140 RETURN
10150 IF A$ <> "S" THEN 10180
10160 M = 32.06
10170 RETURN
10180 IF A$ <> "NA" THEN 10210
10190 M = 22.98977
10200 RETURN
10210 IF A$ <> "BR" THEN 10240
10220 M = 79.904
10230 RETURN
10240 IF A$ <> "I" THEN 10270
10250 M = 126.9045
10260 RETURN
10270 IF A$ <> "P" THEN 10300
10280 M = 30.97375
10290 RETURN
10300 IF A$ <> "MG" THEN 10330
10310 M = 24.305
10320 RETURN
10330 IF A$ <> "AU" THEN 10360
10340 M = 196.967
10350 RETURN
10360 PRINT "I'M SORRY, BUT I DON'T HAVE THIS ELEMENT L
    ISTED"
10370 PRINT "TRY ANOTHER COMPOUND"
10380 END
```

Program Listing

✓ *Reader Service—see page 242*

Have Tandy tally your time cards.

Punch Out

James J. Conroy, KA3FAL
57 E. Garrison St.
Bethlehem, PA 18018

Using the TRS-80 to compute your company's payroll is not new.

Most software designers ignore one of the most time-consuming jobs of payroll computation: the tallying of time cards. I've put together a short program called *Punch Out*, which allows clerks to whiz through the totaling of time cards and the calculating of certain payroll deductions.

The heart of the time card tally section is a looping routine based on the number of time-in and time-out entries which the operator decides to make per employee. My application involves workers punching in and out more than once the same day. I have called such a time-in/time-out entry a "time period".

After the computer displays some brief instructions regarding the data entry format, it prompts the operator by asking "How many time periods are there?".

After the appropriate number is entered, the computer then asks the operator for the time-in (including AM or PM) and the time-out. For those who are used to mentally calculating the hours and minutes, the results will seem like heaven. After each entry, the computer displays the elapsed time between the punched entries in hours and minutes, and in total minutes. After the last time period is entered, the computer pauses and the list of payroll entries for that employee appears in total minutes. This is a check against the initial computations. The total hours and minutes worked will then appear on the screen.

What's Left

I have included the rest of our method for payroll computing in the number lines following 360.

After the display of the total time worked, the computer asks, "What iz da base rate of pay boss?". This system is geared strictly for an hourly rate. Once the appropriate employee rate is entered, the gross pay amount appears on the screen.

The computer then calculates the withholding deductions. The results for the final payroll calculations then appear on the screen.

For our purposes we derived the following four deductions:

- FICA (6.13% of the Gross Pay).
- Federal withholding deduction (This must be entered from Schedule E).
- State Tax (2.2% of the Gross Pay).
- Local Tax (1% of the Gross Pay).

The bottom line of this series of deductions is the Net Pay.

How It Works

The system is based on four subroutines which encompass all the possible combinations of time-in/time-out entries relevant to figuring elapsed work time. If you check the REM statements, you'll see the following procedures:

- The computer splits up the time entries into hour and minute components.
- The computer checks time zones (AM or PM). If they are identical, it goes to line 125, and if they are different, it goes to line 215.
- The computer handles a time punched in on the hour differently than a time punched in at any other minute, since one process is simply a counting of time from the even hour (punched in) and the other is a more complex job of juggling minutes. It does the job you hated: adding and subtracting those stray minutes.

The total time worked figure is accurate, since it is figured from the total minutes worked. You can't be any fairer than to pay an employee for his time worked to the minute! Some time clocks do limit the time-in and time-out to six minute intervals (1/10 of an hour) while others are exact. The program will fit any system.

A final division of the total minutes by 60 yields the total hours and minutes worked, which is then multiplied by the pay rate to produce the gross pay figure.

Program Modifications

1. Few users may want the individual entry, Federal Withholding Deduction from Schedule E. You can certainly use the classical percentage method that allows your computer to figure the correct deduction easily. This is illustrated in Schedule E. Routines are available in standard program books.

2. If you want a written record, change the PRINT state-

ments in lines 315, 320, 325, 375, 390, 395, 400, 405, 410, 415 to LPRINT and you will have your documentation.

3. If you are lucky enough to have an appropriate payroll program complete with data and record files written in BASIC, why not renumber it? You may stick the time card tally routine in the beginning and feed its results to the total time worked inquiry/entry of your program.

4. If you want to work overtime, you can put provisions in to detect and account for it and your overtime pay rate.

You'll find that there is a possible time-in/time-out entry that may cause the program to foul up.

If you have an employee who punches in exactly at 12 o'clock in one time zone and punches out at 12 o'clock in another time zone, the computer will come up with zero hours worked instead of 12. Since the chances of this happening are low, it isn't worth the extra lines to detect this event and provide for it.■

```
4 '    PUNCH - TIME CLOCK PROGRAM BY:
5 '    JAMES J. CONROY - 57 E. GARRISON ST. BETHLEHEM,
   PA 18018
10 CLS:CLEAR
15 PRINT" THIS PROGRAM IS DESIGNED TO PRINT OUT THE TOT
   AL TIME (IN HRS & MINUTES)  AND...IN TOTAL MINUTES
   .... BETWEEN THE TIME PUNCHED IN & THE TIME PUNCHE
   D OUT."
20 PRINT" CAREFULLY ENTER EACH TIME WITHOUT ANY PUNCTUA
   TION (FOR EXAMPLE, ONE O'CLOCK = 100   OR  ELEVEN
   FORTY FIVE = 1145).  ALSO, YOU MUST HIT ENTER AFTE
   R EACH ENTRY!":PRINT
25 INPUT "HOW MANY TIME PERIODS (IN AND OUT) ARE THERE?
   ";Q
27 DIM L(Q)
30 T=0
35 FOR D=0 TO Q:IF D=Q THEN GOTO 330
40 PRINT:INPUT "TYPE THE TIME PUNCHED IN";T1
45 INPUT"NOW ENTER AM OR PM (A/P)";Z1$
50 INPUT "TYPE THE TIME PUNCHED OUT";T2
55 INPUT"NOW ENTER AM OR PM (A/p)";Z2$
60 CLS
65 REM THIS SPLITS TIME IN INTO HOURS AND MINUTES
70 FOR X=100 TO 1200 STEP 100
75 IF T1-X < 60 THEN PRINT X*.01,T1,T1-X,Z1$:GOTO 90
80 NEXT X
85 REM THIS SPLITS TIME OUT INTO HOURS AND MINUTES
90 FOR Y=100 TO 1200 STEP 100
95 IF T2-Y<60 THEN PRINT Y*.01,T2,T2-Y,Z2$:GOTO 110
100 NEXT Y
105 REM THIS TESTS TO SEE IF TIME ZONES ARE SAME-OR NOT
110 IF Z1$=Z2$ THEN GOTO 125
115 IF Z1$<>Z2$ THEN GOTO 215
120 REM THIS TESTS FOR MINUTES
125 IF T1-X>0 THEN 155 ELSE 130
130 REM"THERE ARE NO MINUTES"
135 HRS=Y*.01-X*.01:IF X*.01=12 THEN HRS=HRS+12
140 PRINT"TOTAL TIME WAS";HRS;"HOURS AND";T2-Y,"MINUTES
    "
145 PRINT"OR:";HRS*60+(T2-Y);"TOTAL MINUTES"
150 L(D)=HRS*60+(T2-Y): PRINT L(D);: NEXT D
155 REM"THERE ARE MINUTES"
160 HRS=Y*.01-X*.01-1:IF X*.01=12 THEN HRS=HRS+12
165 MIN=60-(T1-X)
170 REM "HOURS TIL TIME OUT ARE";HRS
175 REM"MINUTES LEFT IN INITIAL PERIOD WERE:"MIN
180 TMIN=MIN+(T2-Y)
185 REM TMIN;"THIS IS THE TOTAL MINUTES LEFT"
190 IF TMIN=>60 THEN TMIN=TMIN-60:HRS=HRS+1
195 PRINT"TOTAL TIME WAS:";HRS;"HOURS AND";TMIN;"MINUTE
    S"
200 PRINT"OR:";HRS*60+TMIN;"TOTAL MINUTES"
205 L(D)=HRS*60+TMIN: PRINT L(D);:NEXT D
210 REM THIS SECTION IS USED WHEN TIME ZONES ARE DIFFER
    ENT
215 IF T1-X>O THEN 275 ELSE 220
220 REM THERE ARE NO MINUTES
225 REM HTT=HOURS TIL TWELVE
230 HTT=12-X*.01
235 REM"HOURS FROM TIME IN TO 12:"HTT
240 REM HFT=HOURS FROM TWELVE TO TIME OUT
245 HFT=Y*.01: IF Y*.01=12 THEN HFT=HFT-12
250 THRS=HTT+HFT
255 PRINT"TOTAL TIME WAS:";THRS;"HOURS AND";T2-Y;"MINUT
    ES"
260 PRINT"OR:";THRS*60+(T2-Y);"TOTAL MINUTES"
265  L(D)=THRS*60+(T2-Y): PRINT L(D);:NEXT D
270 REM THIS SECTION IS IF TIME IN HAS MINUTES
275 HTT=12-X*.01-(1)
280 MIN=60-(T1-X)
285 REM"HOURS TIL TWELVE ARE:";HTT;"MINUTES LEFT";MIN
290 TMIN=(T2-Y)+MIN
295 HFT=Y*.01: IF Y*.01=12 THEN HFT=HFT-12
300 IF TMIN=>60 THEN TMIN=TMIN-60:HFT=HFT+1
305 REM"THE HOURS FROM 12 TIL PUNCHED OUT SHOULD BE";HF
    T
310 THRS=HTT+HFT
315 PRINT"TOTAL TIME WAS";THRS;"HOURS AND";TMIN;"MINUTE
    S"
320 PRINT"OR:";THRS*60+(TMIN);"TOTAL MINUTES"
325 L(D)=THRS*60+(TMIN):  PRINT L(D);: NEXT D
330 FOR R=1 TO 2000:NEXT:CLS: PRINT "NOW WE CAN CALCULA
    TE TOTAL TIME FOR PAY"
335 FOR C=0 TO D-1
340 PRINT L(C):NEXT C
345 FOR C=O TO D-1
350 T=T+L(C):NEXT C
355 PRINT T
360 PRINT"TOTAL TIME WAS";T/60;"HOURS"
365 INPUT"WHAT IZ DA BASE RATE OF PAY, BOSS?";G
370 GP=INT((G*(T/60))*100+.5)/100
375 PRINT"GROSS PAY IS:";:PRINT USING"$$###.##";GP
380 INPUT"PLEASE ENTER THE FED. DEDUCTION (FROM SCHEDUL
    E 'E')";F
385 PRINT"THE DEDUCTIONS ARE:"
390 SOC=INT((GP*.0613)*100+.5)/100:PRINT"1.  FICA (SOC.
    SEC.) IS:";SOC
395 PRINT"2.  FED. WITHHOLDING IS:";F
400 ST=INT((GP*.022)*100+.5)/100:PRINT"3.  STATE TAX IS
    :"ST
405 CT=INT((GP*.01)*100+.5)/100:PRINT"4.  lOCAL TAX IS:
    "CT
410 PRINT"TOTAL DEDUCTIONS ARE";SOC+ST+F+CT
415 PRINT:PRINT"**** THE NET PAY IS:";GP-SOC-F-ST-CT
420 INPUT "DO YOU WANT TO RUN ANOTHER PAY? (Y/N)";A$
425 IF A$="Y" THEN CLS:CLEAR:GOTO 25
430 IF A$="N" THEN CLS: PRINT "WASNT THAT FUN?....BYE!"
435 END
```

Program Listing.

Avoid it with this payment scheduling program.

The Final Notice

Walter J. Atkins, Jr.
Qtrs 4410A
USAF Academy, CO 80840

Nothing is more boring or distasteful to me than making the monthly payments that feed inflation and deplete my checking account. This short program may not make out the checks for you, but it does cut the boredom by identifying which accounts must be paid and when. And no creditor will be overlooked!

This program offers you three reporting choices. You may display all accounts with payments due on the first of the month, all with payments due on the fifteenth of the month, or all with payments due between any other two dates you specify.

The Program

The account information files are stored in data statements beginning at line 6300. The format is:

Line # DATA "account name","account number",due date,amount due

For example:

```
6300   DATA"DOE & CO","A1234-RT",2,
12.50
```

After selecting a method of reporting, the program then asks if the account numbers are to be displayed. A response of "N" suppresses the display of account numbers.

The program next displays the accounts due. If there are more than five, it will display them in output pages of five accounts each. The final output is the total number of accounts due and the total amount of payments due.■

Program Listing 1. Payments Due.

```
100 REM   PAYMENT DUE DATE PROGRAM
200 REM   BY
300 REM   WALTER J. ATKINS, JR.          MAR 79
400 REM   QTRS 4410A USAF ACADEMY, CO 80840
500 REM   (303) 472-1362
600 L=0:H=0:F=0:T=0:C=0:RESTORE
700 CLS:PRINTTAB(20)"PAYMENT DUE DATES"
800 PRINT:PRINT"FUNCTIONS AVAILABLE ARE :"
900 PRINT:PRINT"   1 - ACCOUNTS DUE FIRST OF MONTH
1000 PRINT:PRINT"   2 - ACCOUNTS DUE 15TH OF MONTH
1100 PRINT:PRINT"   3 - ACCOUNTS DUE BETWEEN ANY TWO DA
     TES
1200 PRINT:PRINTTAB(10)"SELECT ONE  ":INPUT"==>"; D:CLS
1300 IF D>3 OR D<1 THEN 700
1400 PRINT:PRINT
1500 PRINT"DO YOU WANT ACCOUNT NUMBERS DISPLAYED (Y OR
     N)";
1600 INPUT I$:I$=LEFT$(I$,1)
1700 IF I$<>"Y" AND I$<>"N" THEN CLS: GOTO1400
1800 CLS
1900 ON D GOTO 2000, 2400, 2800
2000 GOSUB 4700
2100 IF D<17 GOSUB 5000
2200 IF F<>1 THEN 2000
2300 GOSUB 5700:GOTO 600
2400 GOSUB 4700
2500 IF D>=17 GOSUB 5000
2600 IF F<>1 THEN 2400
2700 GOSUB 5700:GOTO 600
2800 PRINT:PRINT
2900 INPUT"START DATE (MAX=31)";D1
3000 IF D1>31 OR D1<1 THEN CLS:GOTO 2900
3100 INPUT"STOP DATE (MAX=31)";D2
3200 IF D2>31 OR D2<1 THEN CLS:GOTO 3100
3300 IF D2<D1 THEN CLS:GOTO 2900
3400 CLS
3500 GOSUB 4700
3600 IF D>=D1 AND D<=D2 GOSUB 5000
3700 IF F<>1 THEN 3500
3800 GOSUB 5700:GOTO 600
3900 PRINTC;".  ";N$;
4000 PRINTTAB(28)"DUE DATE :";D;
4100 PRINTTAB(45)"AMOUNT $";A
4200 IF I$="Y" PRINTTAB(6)"ACCOUNT NUMBER : ";A$
4300 L=L+1
4400 PRINT:IF L=5 THEN PRINT"PRESS ENTER TO CONTINUE ?"
     ;:INPUTY$
4500 IF L=5 THEN CLS:GOSUB6100:L=0
4600 RETURN
4700 READ N$,A$,D,A
4800 IF N$="END"THEN PRINT"NUMBER OF ACCOUNTS:";C,"TOTA
     L DUE $";T:F=1
4900 RETURN
5000 IF N$="END" THEN 5600
5100 C=C+1
5200 T=T+A
5300 IF H=0 GOSUB 6100:H=1
5400 GOSUB 3900
5600 RETURN
5700 PRINT:INPUT"CONTINUE (Y OR N) ";I$:I$=LEFT$(I$,1)
5800 IF I$<>"Y" AND I$<>"N" THEN 5700
5900 IF I$="N" THEN CLS:END
```

Program continues

```
6000 RETURN
6100 PRINTTAB(15)"A C C O U N T S   D U E":PRINT
6200 RETURN
6300 DATA"DOE & CO","A1234-RT",2,12.50
6400 DATA"JONES","22233",5,1423.56
6500 DATA"SMITH","3344E",22,34.45
6600 DATA"TOM","SA22234",25,56.78
6700 DATA"UNIVERSAL INDUSTRIES","123HH4",30,56.77
6800 DATA"HARRY","QW223AS-A",3,34.67
6900 DATA"DICK","ABC 123",14,75.25
7000 DATA"JANE","23-456",26,67.89
7100 DATA"END","",0,0
7200 END
```

Sample Run

```
FUNCTIONS AVAILABLE ARE :
  1—ACCOUNTS DUE FIRST OF MONTH
  2—ACCOUNTS DUE 15TH OF MONTH
  3—ACCOUNTS DUE BETWEEN ANY
    TWO DATES
SELECT ONE
= =>?1

DO YOU WANT ACCOUNT NUMBERS DISPLAYED (Y OR N)?N

                ACCOUNTS DUE
1. DOE & CO              DUE DATE :2    AMOUNT $ 12.50
2. JONES                 DUE DATE :5    AMOUNT $ 1423.56
3. HARRY                 DUE DATE :3    AMOUNT $ 34.67
4. DICK                  DUE DATE :14   AMOUNT $ 75.25
NUMBER OF ACCOUNTS: 4        TOTAL DUE $ 1545.98
CONTINUE (Y OR N)? Y

FUNCTIONS AVAILABLE ARE :
  1—ACCOUNTS DUE FIRST OF MONTH
  2—ACCOUNTS DUE 15TH OF MONTH
  3—ACCOUNTS DUE BETWEEN ANY
    TWO DATES
SELECT ONE
= =>?2

DO YOU WANT ACCOUNT NUMBERS DISPLAYED (Y OR N)?Y

                ACCOUNTS DUE
1. SMITH                 DUE DATE : 22   AMOUNT $ 34.45
   ACCOUNT NUMBER : 3344E
2. TOM                   DUE DATE : 25   AMOUNT $ 56.78
   ACCOUNT NUMBER : SA22234
3. UNIVERSAL INDUSTRIES  DUE DATE : 30   AMOUNT $ 56.77
   ACCOUNT NUMBER : 123HH4
4. JANE                  DUE DATE : 26   AMOUNT $ 67.89
   ACCOUNT NUMBER : 23-456
NUMBER OF ACCOUNTS: 4            TOTAL DUE $ 215.89
CONTINUE (Y OR N)?Y
FUNCTIONS AVAILABLE ARE:
  1—ACCOUNTS DUE FIRST OF MONTH
  2—ACCOUNTS DUE 15TH OF MONTH
  3—ACCOUNTS DUE BETWEEN ANY
    TWO DATES
SELECT ONE
= =>?3

DO YOU WANT ACCOUNT NUMBERS DISPLAYED (Y OR N)?N

START DATE (MAX = 31)?1
STOP DATE (MAX = 31)?5

                ACCOUNTS DUE
1. DOE & CO              DUE DATE : 2   AMOUNT $ 12.50
2. JONES                 DUE DATE : 5   AMOUNT $ 1423.56
3. HARRY                 DUE DATE : 3   AMOUNT $ 34.67
NUMBER OF ACCOUNTS               TOTAL DUE $ 1470.73
CONTINUE (Y OR N)?N

READY
>
```


✓Reader Service—see page 242

An in depth review of the Shack's Disk Editor/Assembler package.

#26-2202

TRS-80 Disk Editor/Assembler
Tandy/Radio Shack
Ft. Worth, TX
$99.95

Guerri F. Stevens
General Business Systems, Inc.
1420 Main St. (Suite 130)
Glastonbury, CT 06033

This article describes the Radio Shack Disk Editor/Assembler package (catalog number 26-2202) written by Microsoft and licensed to Radio Shack. The version reviewed here runs on the TRS-80 Model I.

The package includes features which link together assembly language and FORTRAN programs. It also includes a library of FORTRAN subroutines which can be accessed from assembly language. Those features are beyond the scope of this article.

Background

I have been writing programs in assembly language for the past year. I have been using Radio Shack's cassette Editor/Assembler (EDTASM), modified to run under DOS and to read from and write to disk.

My own technique for designing and using a program is to divide the program into several logical units, which I call modules. I try to keep the modules small so that they are easy to review manually. I like to have each module be a separate source file for two reasons: first, to reduce the possiblility of running out of space in EDTASM's buffer; second, to allow a small section of a program to be reassembled when an error is detected, rather than having to reassemble a large program.

I assemble the modules separately, leaving space at the end of each one for expansion. I then load the generated object code into memory individually for each module and use the DUMP command to write out a single, executable file. This procedure is somewhat tedious and inefficient for a number of reasons. First, a CALL or other access from one module to a location in another requires equate statements to define the external addresses. The equate statements have to be changed whenever locations in the external modules are changed.

Secondly, the space occupied by the program is larger than really necessary due to the gaps left at the end of each module for expansion. Once the program is debugged, modules can be reassembled without the gaps, but that is a time-consuming process, requiring not only reassembly, but also changes to the aforementioned equate statements. The final version has to be tested to make sure no errors were introduced.

What I really wanted was a method of writing the modules and assembling them without regard to any specific memory locations. They could then be linked together consecutively, thus eliminating the need for any gaps. Any one could be reassembled, becoming smaller or larger in the process. Only the linking would have to be repeated to construct a working program again. This would require an assembler which produced object code which could be relocated as desired. The object code would contain information about references to instructions or data external to the module being processed. A linkage editor would be required to read the object code from one or more modules, perform the relocation, and resolve any external references.

Radio Shack's Disk Editor/Assembler package seemed to meet these requirements, so I purchased it. The package comes in a three-ring binder containing the documentation and two diskettes. The package has five components: an editor (EDIT), a macro assembler (M80), a cross-reference processor (CREF80), a linking loader (L80), and a FORTRAN library (FORLIB). The editor, macro assembler, and cross-reference processor are on one diskette; the loader and the FORTRAN library are on the other. Both diskettes contain the DOS system, so they may be used in drive 0.

My approach to using the new system was to read the manual, giving it a superficial study, and noting questions for further investigation. This gave me enough knowledge to try a simple experiment to ensure that the system was operating correctly, at least as far as the basics were concerned. To perform the experiment, I wrote a short program which displayed my name on the screen and returned to DOS. Using the editor, I keyed in the program and saved it on the disk. Then, I assembled it using the macro assembler. At this point, I ran into a snag. A fatal error was detected on one of the instructions and on using the BREAK key to exit from the assembler, the system entered Level II BASIC, with the MEMORY SIZE prompt appearing on the screen! I returned to DOS, corrected the program error, and tried again. All went well this time. I used the loader to load and execute the program, and it worked fine.

Editor (EDIT)

EDIT is a stand-alone program rather than a combination editor and assembler as is EDTASM. The program is invoked by typing EDIT under DOS. It responds by asking for the name of the file to be edited, which is entered in the usual FILESPEC form. To indicate that a new file is being created, you use the BREAK key, rather than the ENTER key after the file specification.

EDIT can be used on files created by it, BASIC files in the ASCII format, and any other files in a suitable format. Some care must be used in editing BASIC files, however. The manual points out some things to avoid.

After the file specification is entered, EDIT displays its title and copyright and the number of free bytes of storage. You proceed with your editing in the usual manner. You will notice that EDIT makes disk accesses from time to time during the editing session; this happens because it doesn't keep the entire file being edited in memory all at once. You may return to DOS by exiting and writing out the edited file, or by quitting and discarding changes (very useful if you've made a dreadful error!).

Features

EDIT has all the capabilities of EDTASM and the BASIC editor although some of the commands are different. EDIT's features include:

- Insertion of new lines and replacement of existing lines,
- Renumbering of lines,
- Displaying all or part of the file on the screen or the line printer,
- Changing of text within a line or adding text,
- Searching the file for a specific string.

EDIT also has some new features and additions to the old ones:

- Line numbers may go all the way up to 99,999, if desired. In additon, the file may be broken into pages, using special commands to insert and delete page marks. Line numbering may start over in each page, thus providing for many more lines or for a printed listing with page breaks at points other than the bottom of a page. A drawback to the page marks, however, is that commands require a page number as well as a line number when page marks are present.
- For insert and replace commands, a temporary line number increment may be specified. This is a nice feature when you want to insert several lines between two others and don't want to affect the numbering scheme for later insertions at the end of the file.
- Ranges may be specified in terms of a number of lines as well as by beginning and ending line numbers. For example, P100:200 displays lines 100 through 200 on the screen. The command P100!10 displays 10 lines on the screen, starting with line 100.
- The BREAK key is echoed on the screen as a $.
- Editing within a line (called 'Intraline Editing' by the manual) allows you to search for a series of characters (text) as well as a single character.
- The FIND command can locate and display more than one occurrence of the desired string with a single command.
- The SUBSTITUTE command provides for changing a given string to another string for any specified range of lines. The strings need not be the same length. This is useful, especially if you want to replace a string throughout the entire file.

EDIT vs EDTASM

Several commands have different names or formats. These are trivial; reading the manual carefully should help you avoid mistakes. I am not going to list the differences.

The immediately obvious difference is that EDIT is a stand-alone program. When you have created or changed a file you must return to DOS and execute a separate program, M80, to perform an assembly. This is somewhat time-consuming if the file is large, as it must be written out with the changes before returning to DOS. Then, if there are assembly errors, you have to edit it again, write again, and reassemble it.

EDIT places an additional space between the line number and the first character in the line. This means that over-flow to the next line of the screen occurs one character sooner than under EDTASM.

The use of the forward arrow (→) to tabulate causes EDIT to insert blanks in the line, rather than the tab character. This means that once you have tabbed, if you want to backspace you must space back over the blanks; a single backspace doesn't return you to the position preceding the tab. I found this irritating.

When the BREAK key is used to exit from the insert mode (insertion of new lines), if it isn't the first character in the line, the line is retained. EDTASM would not have retained the line.

The use of the backspace key (←) in intraline editing *deletes* characters; under EDTASM it simply moves the cursor backwards. You must use a minus sign and the space bar to backspace a character. To backspace 10 characters, for example, you can either issue 10 minus sign space bar commands or −10 and space bar. This is annoying, too, especially since the D command can be used to delete characters in either a forward or backward direction and the backspace key could have been used to move the cursor backwards.

The PRINT command with no range displays 20 successive lines on the screen. Of course, only 16 lines fit on the TRS-80 screen, so you lose the first four. I am planning to disassemble my editor and see if I can fix this.

The use of the shift key and @ simultaneously to freeze the display has been replaced by Control O (shift key, down arrow (↓) and O all together). This is more awkward, and I can't see why the old method wasn't retained. I could never get it to work.

Shortcomings

There are a number of things about the editor which are irritating. I have mentioned some of them above. Most are minor, and I am sure that once I get familiar with the program I will get used to them. Here I want to mention a few more serious things.

First of all, the files constucted by the editor waste disk space. The line numbers are stored as five-byte fields; if the upper limit for line numbering were set at 65535, the line number could fit in two bytes. Following the line number is hexadecimal 89. I'm not sure what its purpose is, if any. Tab characters are replaced by the appropriate number of spaces, as mentioned above. I don't think that's necessary, except possibly to provide for editing all types of files.

I found that a file which was written under the old EDTASM memory-image format and used five granules of space took seven granules under the new format. That's 10 more sectors—a whole track! I found that I quickly used up the space on a diskette.

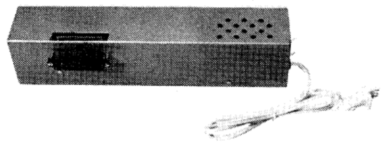
Second, the file handling by EDIT is quite awkward. When you are making changes to an existing file, you must save the new version under a different file specification. This is done so that you have a backup of the original file. It certainly contributes to using up the available disk space. In addition, you now have to develop some system for naming the files so that you know which is the most recent one. Then, when you decide you no longer need the original one, you have to delete it. This may require you to delete *two* files if the editor has created an index file (an index file is created to speed up accesses when the file being edited gets large enough).

Third, in a lengthy editing session, you might want to save your revised version at different points so that a machine or power failure won't require you to repeat all your work. Each time you do this, a *new* file must be created.

Fourth, the default file extension for the macro assembler is MAC. It would be nice if the editor used the same default, but it doesn't. If you forget to specify MAC, you either have to rename the file upon return to DOS or remember to enter the extension when you assemble the program.

TRS-80* MODEL I UPGRADE

If you're tired of waiting for your TRS-80* and need more processing power but don't need the hassels of changing software, the MicroC PCP may be



The 225% Solution.

Simply plug the MicroC PCP into the keyboard expansion port. No installation. No traces to cut. No holes to drill. No wires to solder. Not a mere clock mod but a whole new 4 MHz Z-80A CPU and support circuitry. Programs run reliably more than twice as fast (2.25 times faster) at the SAME CLOCK SPEED AS THE MODEL II. Special proprietary circuitry speeds RAM accesses. Automatic slowdown circuitry slows the processor to normal speed during disk accesses, and during keystrokes to prevent contact bounce on the keyboard.

Available Only from MicroCompatible

✓458

DIMENSIONS: 13" L 3¼" H 2¾" D
COLOR: Grey
POWER REQUIREMENTS: 120 VAC
*Registered Trademark of Tandy Corp.

Send Check for: **$200.00** To:
including shipping and handling

Micro Compatible Inc.
P. O. Box 107
Scaly Mt., N. C. 28775
(704) 526-2782
Office Hours: 8:00 a.m. to 9:00 p.m. EST

LEARN TRS-80® ASSEMBLY LANGUAGE DISK I/O

Your disk system and you can really step out with **REMSOFT's** Educational Module, **REMDISK-1**, a "short course" revealing the details of DISK I/O PROGRAMMING using assembly language.

Using the same format as our extremely popular introduction to assembly language programming, this **"ASSEMBLY LANGUAGE DISK I/O PROGRAMMING"** course includes:

- Two 45-minute lessons on audio cassette.
- A driver program to make your TRS-80® video monitor serve as a blackboard for the instructor.
- A display program for each lesson to provide illustration and reinforcement for what you are hearing.
- A booklet of comprehensive, fully-commented program listings illustrating sequential file I/O, random-access file I/O, and track and sector I/O.
- A diskette with machine-readable source codes for all programs discussed, in both Radio Shack EDTASM and Macro formats.
- Routines to convert from one assembler format to the other.

This course was developed and recorded by Joseph E. Willis, for the student with experience in assembly language programming; it is an intermediate-to advanced-level course. Minimum hardware required is a Model I Level II, 16 K RAM one disk drive system.

REMDISK-1 only $29.95

Dealer inquiries invited

REMsoft, INC.
571 E. 185 St.
Euclid, Ohio 44119
(216) 531-1338

✓70

Includes $1.50 for shipping and handling.
Ohio residents add 5½% sales tax.
TRS-80® is a trademark of the Tandy Corp.

Bayesian Investment Services

Send for free catalog discussing TRS-80 programs available for evaluating:

Options & Futures
Stocks & Bonds
Convertible
& Other Hedges
Portfolio Position
Tax Positions
Market Timing

Write or call:

Bayesian ✓186
Investment
Services
Dept. H3
757 Santa Rosita
Solana Beach, CA 92075
(714) 755-6225

Fantastic New Tool for *Voice Synthesis*

NO EQUAL ON THE MARKET

This peripheral adds high quality communication to the following:

TRS-80
S-100 Bus systems
Other 8 bit processors

The TLP-52 can give:

High quality speech (up to 800 words)
Phonetic speech (unlimited vocabulary)
Software generated speech
Voice recognition
From $200

For more information contact:

The Learning Place
5620-176th S.W. ✓198
Lynnwood, Washington-98036

Fifth, M80 accepts lines of up to 132 characters. EDIT will create and process 255 characters in a line. This could cause a problem, but you are unlikely to create a line longer than 132 characters.

Sixth, the manual states that an INSUFFICIENT MEMORY error can occur under certain conditions. I have not encountered this error yet. However, I think that the editor should have some means of informing the user when memory starts to fill up. The user should have a command available which displays the amount of memory left.

Seventh, there is no way to invoke the editor and process more than one file. If several files must be edited, as is usually the case when making changes to a multiple-module program, the editor must be entered, the first file processed, and then a return to DOS made to invoke the editor again. This is not really serious, just annoying and time-consuming.

Eighth, while in EDIT, there is no way to append another file to the one you are editing. This would be useful. It would be especially nice to be able to create files of macros, for instance, and then insert them in programs as desired (more about macros later).

Finally, there is no way to read and process existing EDTASM source files. I wrote a conversion program which reads the existing files (either in our disk format or in the cassette format) and converts them to the EDIT format. General Business Systems, Inc., Glastonbury, CT will be selling the conversion program.

Macro Assembler (M80)

The macro assembler is the portion of the Disk Editor/Assembler package which converts the source language module into relocatable form. The relocatable module can then be processed by the loader to create an executable program. The assembler is invoked by typing M80 under DOS. When it is ready to accept commands, it displays an asterisk on the screen.

Commands to M80 consist of file names and switches. Three file names, in the DOS file specification format may be given. They are the object (relocatable) file, the listing file, and the source file. The object and listing files may be omitted, in which case no object (relocatable) code or listing will be produced during the assembly. File extensions are optional, and default extensions will be used if they are omitted.

The switches are special instructions to the assembler. They tell it to produce a cross-reference file, list addresses in octal or hexadecimal (the default), assemble 8080 or Z80 mnemonics (Z80 is the default).

Once you enter a command, the assembly proceeds. Errors are displayed on the screen as they are detected. At the end of the assembly, the total number of errors is displayed. You can then assemble another module by entering another command, or exit by hitting the break key.

M80 supports the standard Z80 and 8080 mnemonics, which I will not describe. It has many additional features—so many, in fact, that after two thorough readings of the manual I have still not invented ways of using them all!

To me, the most important feature is the ability of M80 to assemble a module in relocatable form. By this I mean that it can assemble a module without regard to its eventual memory location. It assigns addresses relative to some base address, usually zero. The loader (described later) determines the actual addresses and adjusts instructions within the module accordingly.

The macro assembler has four location counters for assigning addresses. These are the common counter, the absolute counter, the data-relative counter, and the program-(code-) relative counter. The four pseudo-operations COMMON, ASEG, DSEG, and CSEG invoke the counters, respectively, with CSEG being the default. COMMON is provided for use when combining FORTRAN and assembly language programs, and won't be discussed here.

The absolute counter is the same as the original EDTASM counter; statements following it are assembled in absolute, rather than relocatable mode, and will be loaded and executed at the specified addresses. The other two counters provide for relocatability. In fact, if you wish, you may create two separate relocatable program sections.

The implication of the names data-relative and code-relative is that data areas may be separated from instruction areas at the time the program is executed. In fact, either of the two location counters may be used for either data or instructions. The ORG statement may be used with either counter to establish the starting value for the counter. Usually the ORG is left out and a value of zero is assumed.

At this point, you may well ask "What?" Let me try to clarify the concept. Let's take the following example:

```
         LD    A, (PROCSW)
         OR    A
         RET   NZ
         .
         .
         .
PROCSW:  DS
```

Under EDTASM, somewhere preceding this series of instructions would be an ORG statement indicating the starting point for the program. Let's assume that the ORG was for location 7000H and it immediately preceded the LD instruction. This means that the LD instruction would be assembled at 7000, the OR at 7003, and the RET at 7004. PROCSW would be somewhere farther on, depending on what lies between the RET and it. Using M80, you get this same result by including the following statements ahead of the LD:

```
ASEG
ORG    7000H
```

Now let's see what we can do with the other two location counters. Suppose this is just a portion of a program. Suppose further that each module in the program contains both data and instructions, but when the whole thing is combined, you want all the instructions to be together and all the data to be together, following the instructions.

This is what you have to do: Ahead of the LD statement, insert CSEG (if there have been no prior ASEG or DSEG statements in this module, the CSEG is assumed and you can omit it). Ahead of the DS, insert DSEG. All the statements in the module from the CSEG through the last one prior to the DSEG are in the code-relative, or program-relative portion of the module. All the statements following the DSEG are in the data-relative portion of the module.

In the absence of any ORG statements, the LD instruction will be assembled at relative address zero, the OR at relative address 3, and the RET at relative address 4. PROCSW will have relative address 0 also, but in the data-relative portion of the module. When an executable program is created by the loader, a base address is added to the relative addresses from the assembly to determine the actual memory location for each item. If, for example, the base address for the program-relative portion is 5200 (hex), the LD instruction will be at 5200, the OR at 5203, and the RET at 5204. Instructions from other modules will follow. Then, data areas (everything assembled under DSEGs) will follow. If the base address for the data-relative portion is 5500, PROCSW will be at 5500, the next data item at 5501, etc. The operand of the LD instruction, assembled as 0 will be adjusted by the loader to 5500 automatically. Note: Special instructions must be given to the loader to accomplish this particular arrangement of data and instructions.

Frankly, I just assemble everything under CSEGs (assumed anyway by M80). This feature of the assembler allows you to separate instructions and data. The main point is, as mentioned before, that the program is assembled without regard to its final resting place in memory. This means that if you have a program consisting of three modules, and the middle one needs to have something added to it, the other two don't have to be reassembled. The program can be linked together again by the loader after the middle module has been reassembled, and the third module will follow the second one at the proper address.

M80 has several pseudo-operations which facilitate modular programming. The EXTRN or EXT pseudo operation is used to identify symbols which are defined elsewhere. The ENTRY or PUBLIC pseudo-op is used to define symbols which are defined within the module at hand but are to be known outside of it. The loader reconciles these when constructing an executable program.

The macros are another big feature of the macro assembler. A macro is a model; it is a series of instructions which can be placed in a program at several different points by using the name of the macro as an operation code. The macro itself is written and placed in the program somewhere prior to the point at which it will first be used. I always put my macros at the beginning of the program. Within the macro there may be one or more dummy arguments (symbols) which are replaced by other, real symbols when the macro is used.

> *"The program . . . worked fine when executed from DOS using the file written by the loader, but would enter DEBUG when executed under control of the loader."*

To illustrate the macro concept, let's say that various routines within a program are executed which set a switch if an error occurs. At the conclusion of each routine, a return to a calling routine is to be made if the switch has been set. This necessitates testing the switch at several points within the program. If the switch is called PROCSW and if a non-zero value means the switch is set, the routine to test it could look like this:

```
LD    A,(PROCSW)
OR    A
RET   NZ
```

If the test has to made at 10 different places, this series of instructions has to be written ten times. To make this routine into a macro, we enter the following statements in the program ahead of the first point at which the test is to be made:

```
TSTPSW   MACRO
         LD      A,(PROCSW)
         OR      A
         RET     NZ
         ENDM
```

TSTPSW is the name of the macro. MACRO is a pseudo-operation that identifies the statements which follow as a macro model. LD, OR, and RET are the model statements, and ENDM is another pseudo-op which terminates the model.

When the test is to be made, instead of writing the three statements (LD, OR, and RET), write TSTPSW as an operation code. The assembler replaces it with the LD, OR, and RET statements at that point in the program.

The example above is quite simple; there are no changes in the statements as they are included in the program. Let's make the example a little different to illustrate the use of dummy arguments. Let's assume that several different switches have to be tested, but all follow the same rule: A non-zero value calls for a return. The macro would look like this:

```
TESTSW   MACRO   ANYSW
         LD      A,(ANYSW)
         OR      A
```

PROSOFT

441

Box 839 / No. Hollywood, Ca. 91603 / (213) 764-3131

WORD PROCESSING

SUBEDIT, SUBSCRIPT, & PROP offer comprehensive, mainframe quality text preparation for many printers, and support for proportional fonts of the Centronics® 737 and Radio Shack® Line Printer IV. 110 pages of documentation includes tutorial and EZSCRIPT. Among the more than 75 commands and formatting options are:

multiple top and bottom titles
line split, join, and duplicate
global search and change
block move and copy
right-justification
chained text files
table of contents
form letters
underlining
superscripts and subscripts
centering
(some printers)

MININIT is a keyboard driver that tells you when the ROM is doing string compression. It also provides automatic key repeat, lower-case support for your lower-case hardware, intelligent screen print, video routing to printer, and direct entry of arrows, underscores.

GENERAL UTILITIES

XTEND40 converts 35-track diskettes to 40 tracks in 15 seconds without tedious hand-copying of files. Requires your hardware and Operating System to support 40 tracks.

QUICK COMPRESS of in-memory BASIC programs. Allows multi-statement lines, doesn't combine lines, removes blanks and remarks at your option. Processes over 200 lines/second.

PRICES

PROP, SUBEDIT, SUBSCRIPT	$40
MININIT	15
XTEND40	15
Quick Compress	20

Please add $3.00 per order to cover diskette and shipping. Checks O.K., but no credit cards or COD's. Calif. residents add 6% sales tax.

All software includes documentation and distribution on diskette for TRS-80® Model 1's (only) with at least 32K of storage. Please specify Operating System (TRSDOS or NEWDOS) and number of disk drives.

FINANCIAL ANALYSIS

Our Financial Analysis System is menu driven for easy user control. The Case Method approach to documentation shows how each program in the system can be applied to solving typical financial problems including:

- Net Present Value Analysis
- Present value of single deposit
- Present value of annuity
- Future value of single deposit
- Future value of annuity
- Break-Even Analysis
- Growth Stock Valuation
- Bond Analysis
- Days Between Dates
- Amortization Schedules
- Sum-of-the-years depreciation
- Straight line depreciation
- Declining balance depreciation
- Monthly payment to amortize a loan
- Number of payments to amortize a loan
- Balloon Payment
- IRR

Minimum hardware requirements:

TRS-80* 16K, 1 Disk, Level II Basic
Please specify Model I or Model II.

Send check or M.O. for $59.95 to CYBERWARE,
3608 Wildgrove, Arlington, TX 76017

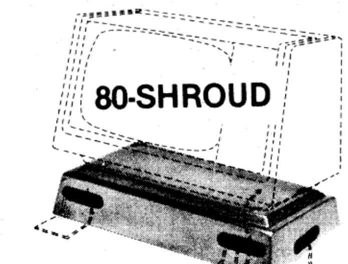
Credit card customers order

TOLL FREE 800-227-1617 ext. 403
California Residents call 800-772-3545 ext. 403

83

Texas residents add 5% sales tax
***TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation.**

cyberware



now you can

RAISE

and tilt your ...
more ...

"80-LNW" VERSION NOW AVAILABLE TO SHROUD THE LNW INTERFACE BOARD
$25.50+$2.00 Shipping & Handling

... silver-gray
...erglass module
...t fits under the
"80" monitor.

$22.50 + $2.00 shipping and handling with check or money order. NYS residents add 7% tax.

SYRACUSE R&D CENTER
Box 125, DeWitt, N.Y. 13214
"our 10th year in R&D" 358

Surplus TRS-80*
RAM Memory Chips

Dealer and Quantity Pricing Available.

DUE TO CHANGES IN PRODUCT STRUCTURE AMERICAN BUSINESS COMPUTERS IS OFFERING SEVERAL THOUSAND 200 NANOSECOND RAM MEMORY CHIPS AT CLOSE-OUT PRICES.

$35 (per 16K set)

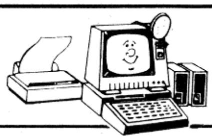
483

AMERICAN BUSINESS COMPUTERS GUARANTEES ALL MEMORY CHIPS TO BE BRAND NEW AND FREE FROM DEFECTS FOR 180 DAYS. PRICE ($45) INCLUDES 8 CHIPS. QUANTITY DISCOUNTS AVAILABLE.

AMERICAN BUSINESS COMPUTERS 397

*TM TANDY CORP.

118 S. MILL ST. - PRYOR, OK 74361 - 918-825-4844



THE MICRO CLINIC 214

DUAL CHARACTER SETS FOR YOUR CENTRONICS 779/RS PRINTER I

Expand the capabilities of your Centronics 779/Radio Shack Printer I with our CLC-1 Lowercase Adapter. The CLC-1 gives you the option of selecting between 2 different UPPERCASE/lowercase character sets so you can choose the print style that is best for your application, all at a fraction of the cost of a new printer. Send for a free sample printout and discover what your Centronics 779/RS Printer I can do for your word processing needs. We think you'll find that its dual character sets, easy installation, and FAST delivery make the CLC-1 an outstanding value!

CLC-1 PRICE: $99 A/T. Includes P/H. CA add 6% tax.
VISA/MC include card number, signature, exp. date, phone number.
MC include interbank number.

master charge — VISA®

THE MICRO CLINIC • 17375 Brookhurst • Suite 114 • Fountain Valley, CA 92708 • (714) 962-0621

```
RET     NZ
ENDM
```

ANYSW is a *dummy* argument. The macro is used as follows, assuming we want to test PROCSW:

```
TESTSW  PROCSW
```

The assembler will insert the following series of statements in places of TESTSW:

```
LD    A,(PROCSW)
OR    A
RET   NZ
```

Note that the dummy argument, ANYSW, has been replaced by the real one, PROCSW.

Macros can be much longer, have many arguments, include test and jump instruction, etc. They should prove to be very useful because of the number of Z80 instructions usually needed to do anything.

M80 has three other functions similiar to the macros. They are all statements which provide for the material between them and the ENDM to be repeated several times. They would be useful, for example, in setting up an array which consists of all zeros. Instead of writing ten DEFB statements, for instance, you could write:

```
REPT  10
DEFB  0
ENDM
```

Ten DEFBs would be inserted in the program at this point. The instructions to be repeated can be made different each time by using the DEFL statement, as shown in the manual. For example, if a table of values is needed, with the values ranging from 0 to 18 in increments of two:

```
X   DEFL   0
    REPT   10
    DB     X; SAME AS DEFB
X   DEFL   X+2; INCREASE VALUE OF
           ;X BY 2
    ENDM
```

The initial DEFL statement makes X equal to zero. The first time the DB is inserted in the program, it will be

```
DB      0
```

The second DEFL makes X equal to 2 (0 + 2 = 2). The second DB generated will be:

```
DB      2
```

Then, X is increased to 4, and so forth.

The other two repeat statements are similar to REPT shown here.

Three pseudo-operations are available for controlling the listing produced by the assembler. The PAGE operation causes the listing to skip to a new page. This is handy if you like to have each major program section start on a new page. The TITLE and SUBTTL operations allow you to specify headings to be printed at the top of each page of the listing.

M80 can produce a cross-reference listing, if you wish. To do so, at the time of the assembly, the C switch must be appended to the source file name in the command. This tells the assembler to produce a cross-reference file rather than a listing file. When the assembly is completed, you have to return to DOS and execute the program CREF80. This program processes the cross-reference file and creates a listing file. The listing file has the cross-reference listing at the end, showing all symbols in alphabetical order, with all references to them.

M80 vs EDTASM

M80 is a stand-alone program. This is a disadvantage, since when assembly errors are detected, you must exit from M80, execute EDIT to correct the errors, then execute M80 again to reassemble the program.

Labels must be followed by a colon. The source conversion program generates the colon automatically after a label.

The listing no longer appears on the screen as the program is being assembled. A separate listing file is produced, which can be displayed upon return to DOS. The LIST command displays it on the screen; the PRINT command displays it on the line printer. Errors detected during assembly are shown on the screen as they are found. When the assembly is completed, the total number of errors is displayed.

The assembler takes two passes to complete its work. Errors are displayed on both passes. Sometimes the same error will cause different error messages on the two passes or will cause errors in other statements on the second pass.

The assembly is somewhat slower that the EDTASM assembly because of the disk accesses. The source program is read from disk, and both the listing and object files are written on disk.

Addresses shown on the listing are reversed from the normal Z80 format. Instead of listing the *least* significant byte first, the assembly lists the *most* significant byte first.

The manual could stand some improvement. I think that a person with little or no assembly language experience would have trouble with it.

Macros are only defined internally; that is, within the program containing them. I think it would be handy to be able to set up a file of macros which could be searched during the assembly process. This way, once you wrote a macro for some general function you could use it in other programs without having to enter it in each one.

The listing uses more space than necessary. It also occupies more space on the disk.

The assembler generates successive line feed (0Ds) in some places. The Radio Shack 1152 printer (Centronics model 779) doesn't recognize a line feed if there's no data in the buffer. This means that the line count maintained by the software doesn't agree with the number of lines printed. Subtitles don't appear on the first page of the listing.

There is an error called a phase error (P error code). It occurs when a label has a different address on the second pass from the address on the first pass. But there's no real explanation as to *why*.

If errors are detected during an assembly, the use of the BREAK key to return to DOS is unpredictable. Usually, in my case, it results in a transfer to Level II BASIC, with the MEMORY SIZE question appearing on the screen. At the nearest Radio Shack computer store, the BREAK key put the system into the 32-character mode and hung up!

Linking Loader (L80)

The loader is the final program in the Disk Editor/Assembler package. Its purpose is to construct an executable program from one or more relocatable object files created by the assembler. The loader is executed by typing L80 from DOS. When it is ready to receive commands, an asterisk is displayed. Commands consist of filenames and/or switches. Multiple names and switches may be combined into a single command by inserting commas between them. The file names used in a command tell the loader what files are to be loaded into memory. In the absence of instructions to the contrary, files are loaded into locations one after another. The switches are, in reality, special commands to the loader. They are two-character commands, consisting of a dash (–) and a letter. They may be presented alone in the commands or appended to a file name.

When a command has been executed, the loader displays the lowest address and the next available address of the program at that point and all symbols unknown (unresolved) thus far. The unresolved symbols may be resolved by loading the files containing them or by searching the FORTRAN library. The FORTRAN library contains various arithmetic functions and conversion functions which may be accessed from assembly language. These are not covered by this article. Once the loading is completed, the program may be saved on disk by the loader, or executed immediately. If there are any remaining unresolved symbols at this point, the loader will automatically search the FORTRAN library for them.

Features

● Relocation: As mentioned in the section on the macro assembler, you may create programs which are assembled with non-absolute addresses. The loader will construct an executable program by moving the relocatable output from the assembler into an available section of memory, and adjusting instruction operands as necessary. The first available memory address is used as a base, and the relative addresses within the module being processed are added to it to obtain absolute addresses. In the absence of instructions to the contrary, the loader will place the first module named at hex address 5200; subsequent modules will follow the first one.

● Resolution of external references: Each time the loader encounters a symbol named in an EXTRN statement, it will search for the same symbol in a PUBLIC or ENTRY statement. When it is found, instructions containing references to the symbol can be adjusted to contain the correct address. When the loader executes a command which causes files to be loaded, it lists all unresolved external references which remain after the loading is completed.

● Separation of instructions and data: If you have used the DSEG pseudo-operation in your program, the loader will place items assembled under its control ahead of items assembled with the CSEG operation. In the absence of loader commands to the contrary, this is done on a module-by-module basis. For example, if you load modules A, B, and C, each containing both DSEG and CSEG areas, memory after loading will look like this:

A's DSEG areas
A's CSEG areas
B's DSEG areas
B's CSEG areas
C's DSEG areas
C's CSEG areas

● Loading at specified addresses: As mentioned above, the loader will put the first named module at hex location 5200. If you wish, you may request that loading begin at some other address. This is done by using two address switches, D and P. The D switch applies to data areas (DSEG) and the P switch to program areas (CSEG). The use of these switches is somewhat confusing, because although the manual states that CSEG is assumed when nothing is specified, the loader seems to assume DSEG; in other words, if you use neither DSEG nor CSEG in your program and specify no address to the loader, it will call your program DATA.

In addition, you must be careful when using the address switches if you have both data and program areas in several modules. The loader will put data from successive modules into contiguous data areas and program information into contiguous program areas. You must be careful to keep the data areas from overlapping the program areas!

● Restart: If you make a mistake, a special switch allows you to reset the loader and start over. This is useful if you inadvertantly loaded the wrong file. You must reload everything once this command is used, however, as the loader forgets what it has done.

● Construction of a disk file: The loader can create a command-format disk file from the loaded program. The default extension for the file is CMD, so that it can be executed from DOS by typing its name.

● Memory map: You can request the loader to display a map on the screen. The map will show the symbols which are defined and their actual load addresses. (Note: this only applies to symbols you have designated as PUBLIC in your program; all others are unknown to the loader). All undefined symbols are listed with an asterisk following them. A different command requests the loader to list only the undefined symbols.

● Execution after loading: You may ask the loader to execute your program immediately, without returning to DOS. If so, the loader first searches the FORTRAN library for any remaining unresolved symbols. It then displays the starting address of the program, the next available memory location, and the message BEGIN EXECUTION. The program is then executed.

Shortcomings

Immediate execution sometimes failed when I tried it. I have not yet determined the exact reason for the failure. The program in question worked fine when executed from DOS using the file written by the loader, but would enter DEBUG when executed under control of the loader.

The display of data and program areas seems backwards in some ways. Since, in the absence of anything else, CSEG is the assumed option in the source program, it seems that the loader should print the word PROGRAM rather than DATA when loading the program.

"The manual could stand some improvement. . . a person with little or no assembly language experience would have trouble with it."

When you specify a starting address for data you must also specify one for the program if you are loading several modules, each of which has both program and data areas. If only the data address is specified, the data area for the first module loaded will start at that address. It will be immediately followed by the program area for the first module. Then, when the second module is loaded, its data area will be put at the end of the data area for the first module, thus destroying some or all of the program area for the first module. No message or warning is given when this happens.

The loader doesn't give you the option of printing a memory map rather than, or in addition to, displaying it on the screen. A printed map is a necessity when trying to debug a program. It's a nuisance to have to write down the addresses manually.

The map displayed by the loader uses only the symbol names, and doesn't include the module names themselves. It would be nice to list the low and high addresses for each module, by name, together with a list of the public symbols within the module and their addresses. I have now developed a convention for symbols (those which are labels on instructions): the first few characters are the same as the module name itself.

The symbols displayed in the map are in the order in which they are encountered, rather than alphabetically.

There is no provision for accepting commands from a source other than the keyboard. When a multi-module program is being tested, it will have to be loaded many times. It would be nice to be able to put the load commands into a disk file and have the loader access them. You can use the .REQUEST psuedo operation in the source files to achieve a similar effect. However, the loading of the requested files doesn't occur until either an exit from the loader or execution occurs.

Summary

There are some improvements I would like to see made to the package. The main ones are:

● Reduce the amount of space required by source and listing files.

● Change the file handling in the editor so that new files don't necessarily have to be created each time changes are made. Perhaps the retention of a backup copy of the original file could be an option.

● Revise the manual. The section on the editor is good, but the assembler and loader sections need work.

I am happy with the package's features, especially the linking-loader and the modular programming features of the assembler. ■

A latter program matter formatter.

LPRINT Formatter II

Charles Z. Tzinberg
6318 Hentage Station Rd.
Belleville, IL 62223

My thanks go to E. M. McCormick for his program "LPRINT Formatter," (*80 Microcomputing*, February, 1980). I have seen several programs which print paged listings in both BASIC and machine language. They require a program to be saved on disk as an ASCII file.

If, like me, you pack your programs tightly by combining lines whenever possible, then trying to load a program saved in ASCII results in that dreaded "direct statement in file" error message.

For those of you who have this problem, or who don't have a disk drive, this program provides the solution.

McCormick's original program was designed specifically for teletype applications, but this version should be generally applicable. I wanted *LPRINT Formatter II* to be usable on *any* TRS-80, disk or cassette based.

Solution

If you have a disk, type the program and save it as an ASCII file (SAVE "LPRINT",A). Don't worry, none of the lines are long enough to cause a direct statement in file error. To use, load the program to be listed, then MERGE "LPRINT", and RUN. You should never use line numbers lower than 19, or when you merge LPRINT these lines will be overwritten. If you use a cassette, use one of the many programs available to merge two programs.

Line two is necessary when operating with 48K, since to PEEK or POKE at addresses over 32767 you must subtract 65536 from the address. This has the unfortunate effect of slowing the program considerably, but is needed nonetheless.

I allowed for any maximum line length input from 64 to 131, with a default value of 64. My program is basically designed for use with a TRS-80 lineprinter (equivalent to a Centronics 779) which allows manipulation of the variable print width to make your listings visually appealing. In line 4 LINEINPUT is used so that commas may be inserted in your header, if desired. The header is limited to 16 spaces less than the maximum line length so it may be properly spaced. Using "IF L0<1 THEN L0 = 19" allows you to enter 1 to list the LPRINT Formatter II program, to use any other line number to start at that line, or to hit ENTER to default to the start of the program immediately following LPRINT Formatter II.

In line five, if the HI line number is found to be less than the L0 line number (which would occur with an entry error or if the default value was invoked by hitting ENTER), HI is set equal to 65529, which is the highest line number allowed in BASIC.

I did not include a line to put a string of dashes at the end of each page, to facilitate cutting if you use roll paper. You need alter only two lines if you want this addition. In place of lines 5 and 15 insert the following:

```
5    INPUT"HI ";HI:CLS:LPRINTSTRING$
     (ML,"–"):LPRINTSTRING$(4,138):IF
     HI<L0THENHI = 65529
15   LPRINTSTRING$(59 – LN, 138):LPRI
     NTSTRING$(ML,"–"):LPRINTSTRIN
     G$(4,138):E = S:PG = PG + 1:GOTO6
```

Any time a program statement is too long to fit on one line, it will automatically wrap around, leaving a space at the start of the line the same length as the line number.

Whenever there is an ELSE in your program it will be printed with a colon preceding it (:ELSE) automatically supplied by the interpreter.

I am sure you will find many uses for LPRINT Formatter II. It uses only 1424 bytes (less than McCormick's "LPRINT Formatter") and, although it does slow down your printer significantly, it prints professional looking program listings. ■

```
1 CLEAR200:DEFINTP:DEFSTRA,D:DIMD(124):P1=32767:K=65536
     :PK=256: E=PEEK(16548)+PK*PEEK(16549):PG=1:GOTO3
2 IFX>P1THENX=X-K:RETURNELSERETURN
3 CLS:READD(1):IFD(1)<>"END"THEN3ELSEREADD(2):IFD(2)<>"
     FOR"THEN3ELSEFORPC=3TO124:READD(PC):NEXTPC:INPUT"L
     INE LENGTH ";ML:IFML<64ORML>131THENML=64
4 X=0:INPUT"ENTER 9 TO TEST PRINT ";X:IFX=9THENLPRINT"X
     ";STRING$(ML-2,32);"X":GOTO4ELSELINEINPUT"HEADER "
     ;A:IFLEN(A)>ML-16THEN4ELSEINPUT"LO ";LO:IFLO<1THEN
     LO=19
5 INPUT"HI ";HI:CLS:IFHI<LOTHENHI=65529
6 LN=0:KL=INT((ML-LEN(A$))/2):LPRINTSTRING$(KL,32);A;ST
     RING$(KL-7,32);"PAGE";:LPRINTUSING"###";PG:LPRINTC
     HR$(138)
7 X=E:GOSUB2:S=PEEK(X):X=E+1:GOSUB2:S=S+PK*PEEK(X):X=E+
     2:GOSUB2:L=PEEK(X):X=E+3:GOSUB2:L=L+PK*PEEK(X):IFL
     >HIORS=0THENENDELSEIFL<LOTHEN14ELSELN=LN+1:PC=LEN(
     STR$(L)):LPRINTL;:FORM=E+4TOS-2:X=M:GOSUB2:M1=PEEK
     (X):IFM1<127ORM1>251THEN10
8 PC=PC+LEN(D(M1-127)):IFPC<MLTHEN9ELSELPRINT" ":PC=LEN
     (STR$(L))+1:LPRINTSTRING$(PC,32);:PC=PC+LEN(D(M1-1
     27)):LN=LN+1
9 LPRINTD(M1-127);:GOTO13
10 IFM1=10THEN11ELSEPC=PC+1:IFPC<MLTHEN12ELSELPRINT" "
11 PC=LEN(STR$(L))+1:LPRINTSTRING$(PC,32);:LN=LN+1
12 LPRINTCHR$(M1);
13 NEXTM:LPRINT" ":IFLN=>57THEN15
14 E=S:GOTO7
15 LPRINTSTRING$(64-LN,138):E=S:PG=PG+1:GOTO6
16 DATAEND,FOR,RESET,SET,CLS,CMD,RANDOM,NEXT,DATA,INPUT
     ,DIM,READ,LET,GOTO,RUN,IF,RESTORE,GOSUB,RETURN,REM
     ,STOP,ELSE,TRON,TROFF,DEFSTR,DEFINT,DEFSNG,DEFDBL,
     LINE,EDIT,ERROR,RESUME,OUT,ON,OPEN,FIELD,GET,PUT,C
     LOSE,LOAD,MERGE,NAME,KILL,LSET
17 DATARSET,SAVE,SYSTEM,LPRINT,DEF,POKE,PRINT,CONT,LIST
     ,LLIST,DELETE,AUTO,CLEAR,CLOAD,CSAVE,NEW,TAB(,TO,F
     N,USING,VARPTR,USR,ERL,ERR,STRING$,INSTR,POINT,TIM
     E$,MEM,INKEY$,THEN,NOT,STEP,+,-,*,/,(,AND,OR,>,=,<
     ,SGN,INT,ABS
18 DATAFRE,INP,POS,SQR,RND,LOG,EXP,COS,SIN,TAN,ATN,PEEK
     ,CVI,CVS,CVD,EOF,LOC,LOF,MKI$,MKS$,MKD$,CINT,CSNG,
     CDBL,FIX,LEN,STR$,VAL,ASC,CHR$,LEFT$,RIGHT$,MID$,"
     "
```

Program Listing

Review of some elementary considerations about computing.

Some Fundamentals

Steve E. Tune, Pres. HMCT
5313 Bissonnet
Bellaire, TX 77401

The TRS-80 Model I can be a very effective business computer, especially considering that a comparable computer bought in 1970 would have cost more than $100,000.

Let's examine the factors that affect any small business computer installation:

- Environment
- Hardware Installed
- Operating System
- Applications Software

Environment

Power noise and surges can affect the memory data of any computer. Therefore careful engineering has gone into providing pure power to most systems. Since most of the micros have very little electrical protection it's no wonder that when your Xerox machine is turned on your system becomes somewhat erratic. You need to protect your computer in two ways. First, you need to isolate it against line transients, spikes, and voltage variations. Second, you need to eliminate RF interference through the line. My recommendation is to use a 250VA minimum Sola isolation transformer (or equivalent) with Corcom RFI filters incorporated in the lines. This package seems to work quite well in some very busy electrical environments.

Hardware Installed

The major flaw in the TRS-80 hardware has been the expansion interface (E/I). The problems with the E/I have been noise on the data lines affecting data in memory, and poor disk I/O. Radio Shack has just introduced a new E/I board that has incorporated into it all the previous fixes and engineering changes. This new board should perform much better than the old model boards. If you have an old E/I board, Radio Shack will exchange a new one for $200 and your old board.

Disk I/O is caused by a different problem. Radio Shack chose Western Digital's 1771 Disk Controller to use in their E/I. This chip comes equipped with a built-in data separator circuit. Most other manufacturers recommend an external data separator be used with the 1771 disk controller.

Radio Shack elected to install the built-in data separator only in the 1771 IC. This leads to somewhat sporadic disk I/O, especially when utilizing the higher tracks.

The solution to this problem is a little board called the Separator, manufactured by Percom Data. The Separator is plugged between your disk controller and E/I. It's available from Percom Data or Houston Micro-Computer Technologies, Inc.

One other problem you may face with your E/I is cheap RAM (Random Access Memory). We have found only two sources of RAM that work consistently in the E/I: NEC and Motorola. These are the same RAM kits that Tandy furnishes.

One other comment about hardware. The six major brands of mini-drives for the TRS-80 are Shugart (used by Tandy), Pertec, Tandon (Tandy), Wangco, MPI and Percom. The Pertec, Wangco, MPI and Percom can all write up to 40 tracks versus Shugart's maximum of 35. Percom will soon be releasing a dual density controller that plugs between your E/I and your disk drive, doubling the capacity of your existing system.

Operating Systems

Many of us have experienced erratic behavior in TRS-80 operating systems. Apparat has since introduced an alternate operating system to TRSDOS, called NEWDOS. With NEWDOS, there are no more disk I/O errors, parity errors during read, key bounce or any of the other problems we had come to associate with TRSDOS. NEWDOS has made the TRS-80 a viable computer system.

Application Software

When purchasing software, it is important to know your supplier rather than shopping primarily by cost. There seems to be a direct correlation between cost and quality in the software market.

A tremendous amount of software has been written around the TRS-80, and, unfortunately, most of it is poor. Nine times out of ten the major shortcoming is not in the code of the program, but how well it interacts with the operator, by means of prompting, extensive error trapping, and adequate documentation. Most manufacturers' software falls in a mediocre category, because most manufacturers offer software only to sell their hardware. In most cases this means the software you purchase may not be the easiest to use or the most flexible to operate.

In the future, better software packages should be available from a variety of producers specifically manufacturing software.■

LEARNING TOOLS

- Spelling Primer
- Vocabulary Builder
- Crossword Generator

EACH PROGRAM FEATURES:

- HUNDREDS OF WORDS
- GRAPHIC SCORING MONITOR
- PROGRAMMABLE SPEED CONTROL

Each Program Available in 4 Levels:

Grades 1-2; 3-4; 5-6; 7-8.
(Specify Grade)

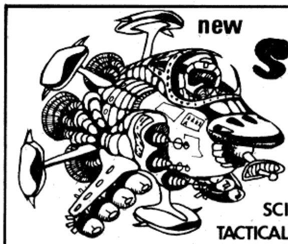
16K - LEVEL II

$4.95 each — any two for **$7.95**

DYNATEK INFO. SYSTEMS

586 CONCORD AVE.
WILLISTON PARK, N.Y. 11596

✓253 *"Quality Software at Affordable Prices"*



new

SPACE ACE 21

SCIFI GAME OF
TACTICAL SPACE COMBAT

Design your own space fighter. You choose the weapons, equipment, engines, and more. Then engage in a one on one sudden death fly-off. Human opponent or 10 different computer opponents. Fight in 2 or 3 dimensions. Action Sounds and Graphics. The most imaginative scifi game we offer.
(written by former USAF phantom pilot) $19.95

More Super Games with Action Sounds and Graphics:

!!! ANTS !!!	Amazing battle between ant colonies. Easy to play. Challenging to master!	$14.95
PARSECTOR V	The ultimate space war. Unique split video, each player has a private display.	$19.95
PARSECTOR 8	Tournament version of Parsector V. Larger galaxy to conquer.	$20.95
HIGH SPEED LIFE	The fastest, most advanced life game. Up to 2000 gen/min possible.	$14.95
NAME THAT STATE QUIZ	Educational fun. Draws State and asks questions.	$14.95

TRS 80 Level II 16k, Model I or III required.
For cassette and instructions send CHECK or MONEY ORDER.

Synergistic Solar Inc. PO Box 560595 Miami FL 33156

Please write for more info. Dealer inquiries invited.

Introducing

mce T.M.
PROGRAMS

Micro-Computer Educational Programs

Interpretive Education, providing leadership in educational programs for basic living skills, introduces a breakthrough series of new Micro Computer Educational (MCE) programs.

The MCE programs are unique in that they offer automatic branching to individual reading levels, variability in vocabulary levels and are educationally sound.

The new programs are cooperatively designed by a combined team of educators and micro computer specialists. Each program is currently designed for application on Apple II* and TRS-80**.

Please call collect today for more information on how MCE programs can aid your teaching efforts with special needs audiences.

*A trademark of Apple Computer, Inc.
**A trademark of Tandy Corporation

For free information and catalog, write or call collect: **(616) 345-8681**

INTERPRETIVE EDUCATION ®
Dept 18G ✓295
2306 Winters Dr. Kalamazoo, MI 49002

SOFTWARE * * CRITIC

PROGRAM REVIEWS FOR THE "80'S" *

STOP wasting your software dollars. START getting timely, complete, understandable, objective, and critical software reviews. Subscribe NOW to the SOFTWARE CRITIC. Published bi-monthly, charter subscription only $9 per year. (Regular $15.00) Canada and Mexico $18.00, others $24.00. Unused portion refunded if you're not satisfied. Send check or money order for $9 to:

NEW!

SOFTWARE CRITIC
P. O. BOX 3CH
UNIVERSITY PARK,
NEW MEXICO 88003

$9/YR

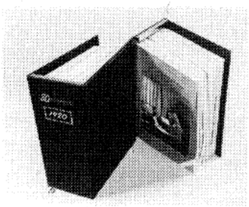
*Currently MODS I & III ✓220

---

80 microcomputing T.M.

BINDERS

order yours today



Keep your library of 80 Microcomputing safe from loss or damage in these handsomely appointed binders with rich dark green covers and gold lettering. Each binder holds 12 issues making an EXCELLENT REFERENCE HANDBOOK. Several binders form a quality library you can be proud of.

$7.50 each . . . 3 for $21.75 . . . 6 for $42.00 . . .
Postage paid in USA. Foreign orders please include $2.50 for postage.

Specify 1980 or 1981

Send check or money order only to:
80 MICROCOMPUTING BINDERS
P.O. Box 5120, Phila., PA 19141

Allow 6-8 weeks for delivery. Please no C.O.D. orders, no phone orders.

---

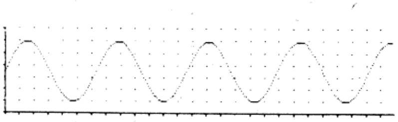
96 CHAR. PER LINE. ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghi

64 CHAR. PER LINE. ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_

80 CHAR. PER LINE. ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmn

BASE II for TRS-80

(The Graphic Line Printer)



64, 72, 80, 96, 120, and 132 Characters per line.
TRACTOR FEED - AUXILIARY CHARACTER FONT - 2K BUFFER

Item	Expansion Interface With	Expansion Interface Without
Model 800 MST	$699.00	$699.00
Printer/Interface Cable	$40.00	-
Modified RADIO SHACK Line Printer II Interface Cable.	-	$90.00
TOTAL	$739.00	$789.00
OUR PRICE	$710.00	$730.00

NOTE: THIS ENTIRE AD HAS BEEN CREATED ON THE BASE II LINE PRINTER.

AVS AUDIO/VIDEO SYSTEMS ✓146
2485 Autumnvale Dr.
San Jose, CA 95132

PH. 408-946-1265

132 CHAR. PER LINE. ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

DISASSEMBLED HANDBOOK FOR TRS-80

VOLUME 3—$18. POSTPAID

Chapter 1: Writing Disassembler Programs
Chapter 2: High Speed Disassemblers
Chapter 3: Spooling Theory & Practice
Chapter 4: Port Encoders & Decoders
Chapter 5: Writing Interrupt Programs
Chapter 6: D/A Converters & Construction
Chapter 7: A/D Converters & Construction
Chapter 8: High Speed Morse Code Program
Chapter 9: Comm. Bulletin Board Systems
Chapter 10: Radio Teletype From A to Z
Chapter 11: Self-Programmed Learning Q/A
Appendix A: Volumes 1-2-3 combined index
Appendix B: Vols. 1-2-3 Pgms. on Disk $20

VOLUME 1—$10. POSTPAID

6th printing

VOLUME 2—$15. POSTPAID

4th printing

—GERMAN & FRENCH LANGUAGE EDITIONS—

RICHCRAFT ENGINEERING LTD.
Drawer 1065, Wahmeda Industrial Park
Chautauqua, New York 14722
phone (716) 753-2654 for COD orders
(US funds: add $4.50 overseas airmail) ✓276

✓ *Reader Service—see page 242*

Fast Eddie says, "Save time and effort. Juggle those utilities around!"

Fast Edit

Dwight K. Illk
103 Tices Lane
East Brunswick, NJ 08816

Do you ever tire of reloading the Editor/Assembler when creating machine language programs? If so, then FASTEDIT is for you.

One Big Drawback

The Editor/Assembler (EDTASM) program available from Radio Shack is a fine piece of software, but it has one big drawback when used on a system without disk: It takes a tremendous amount of time to load the assembler and the source program each time you want to make a minor program change.

EDTASM resides from 4300H to 5D40H in memory, an area also used by BASIC, so the B exit command starts an initialization procedure which destroys part of the EDTASM program. Because of this memory allocation, you must reload EDTASM every time you exit for a trial run of your program. If you use T-BUG to run and debug your program, things get worse! T-BUG lives from 4380H to 4824H, right on top of EDTASM, so when you load one, it clobbers the other.

Finally, when you reenter EDTASM at the starting address of 468AH (18058 decimal), the source program in the text buffer is lost, and the text must be reloaded from tape before starting revisions.

These problems may not be too serious if you never make errors, but for a beginner like myself they are almost enough to make me throw EDTASM back in a drawer. Buying a disk system is one solution, so that loading and saving programs takes almost no time at all. But before going out and blowing a kilobuck on a disk you can't afford and don't really need—read on.

I've found a cure for most of these woes. It trims the time spent dumping and loading tapes by an incredible 90 percent and puts a lot more joy into assembly programming.

Move T-BUG

The first thing to do is remove the T-BUG/EDTASM conflict. Move T-BUG to an unoccupied place in high memory. (See "Get T-BUG High" in the January, 1980, issue of *80 Microcomputing*.) As a precaution, leave room at the top of memory where EDTASM will put the symbol table. I located T-BUG from 7680H to 7B24H which leaves plenty of symbol table room under 7FFFH, the end of 16K.

The trick is to get back and forth from EDTASM and T-BUG without going through BASIC. Use T-BUG high to load EDTASM, and then use the memory command to look at locations 4930H and 4931H. These locations contain the address to which EDTASM jumps when the BASIC command is given. It is now 0000H, the BASIC powerup routine. Modify these locations to the entry point of your relocated T-BUG monitor. Remember to put the most significant bit first. In my case, the monitor entry point is 76A0H, so I put A0H in 4930H and 76H in 4931H. Now jump to EDTASM from T-BUG with a J 468A.

Test your work by replying to the * prompt with a B ENTER and if all is well, you'll be back in T-BUG high! Save your modified version of EDTASM with a P 4300 5D40 468A EDTASM.

Now you can create a program with EDTASM, assemble it onto tape, exit to T-BUG, load the tape, run it, debug it, and return to EDTASM with a J 468A—all without reloading EDTASM or T-BUG!

It's unfortunate when we're in EDTASM and modifying a program that we must reload the text from tape. The solution is: Instead of returning to EDTASM at 468AH, return with a J 4932 command immediately followed by the break key. You'll get the * prompt and your old program will be right there, ready to revise.

Resetting

This sounds pretty good, right? We all know, however, that sooner or later the program we're trying to debug is going to "hang," and the only way out is to hit the reset button. Among other things, the BASIC cold start routine will write garbage into locations 4330H to 434BH making EDTASM useless. But fear not if you are prepared with the program shown in Program Listing 1 (FIXEDT).

Unless your program went berserk and destroyed part of ETDASM or T-BUG, loading this program with the system command and executing it will fix the damage caused by the reset, and will jump to the EDTASM entry location.

Unfortunately, this type of restart loses the program stored

Tape Operation	Normal Time	FASTEDIT Time
Reloading EDTASM	2 1/2 minutes	NONE
Reloading T-BUG	50 seconds	NONE
Reloading Text	Depends on program	NONE
Reset Recovery	2 1/2 minutes	25 seconds with FIXEDT

Note: Only 90 percent time savings are claimed because you still have to save your text buffer from time to time and record/load your machine language code using tape.

Table 1. FASTEDIT Time Savings.

in the text buffer, so make use of the EDTASM W command from time to time so that by reloading the last text buffer tape you will get back on line without much work lost.

The FIXEDT program is completely relocatable and can be put anywhere in memory. It must be high enough so that it doesn't interfere with the text buffer. This extends from the end of the EDTASM up, depending on the length of your program. Also, it shouldn't interfere with your monitor (T-BUG high), or the EDTASM symbol table area at the end of high memory. I start it at 7C90H.

Create the FIXEDT program using the T-BUG M command to POKE it into memory. Save it on tape with a P 7C90 7CC0 7C90 FIXEDT, or other appropriate command if you wish to locate it elsewhere. Now, when it is necessary to use the reset button to get back control of the computer, load the FIXEDT tape using the SYSTEM command followed by a /ENTER to get back to the repaired EDTASM. If FIXEDT is already loaded, simply use SYSTEM and /31888 to do the same job.

The time saved using this system is shown in Table 1. Try it, you'll like it! ■

ADDR	CODE	ADDR	CODE	ADDR	CODE
7C90	11	7CA0	ED	7CB0	18
7C91	30	7CA1	B0	7CB1	04
7C92	43	7CA2	C3	7CB2	E5
7C93	CD	7CA3	8A	7CB3	21
7C94	88	7CA4	46	7CB4	04
7C95	13	7CA5	49	7CB5	FF
7C96	3B	7CA6	43	7CB6	CD
7C97	3B	7CA7	06	7CB7	49
7C98	E1	7CA8	5C	7CB8	43
7C99	01	7CA9	10	7CB9	E1
7C9A	0F	7CAA	FE	7CBA	C9
7C9B	00	7CAB	C9	7CBB	21
7C9C	09	7CAC	E5	7CBC	00
7C9D	01	7CAD	21	7CBD	FF
7C9E	1C	7CAE	00	7CBE	3A
7C9F	00	7CAF	FB	7CBF	28
				7CC0	40

Program Listing 1. FIXEDT.

A look into the data storage capabilities of stringies.

Floppy Tales

Eric Keener WB6EST
9163 Cody St.
Broomfield, CO 80020

The Exatron Stringy Floppy is a mass storage subsystem which attaches to the TRS-80.

The goal of this article is not to describe the Stringy, but, instead, its capabilities to store data. It's about the Stringy Floppy Phonebook. This is called a phonebook but also refers to any other similar data stored on a floppy wafer.

This program could be converted for use with a disk system, although I will not offer any methods because I'm not familiar with them. As for cassette tapes, as you can see in lines 110 and 900, the program can be changed for them. However, the slow speed makes it impractical.

Just how fast is this Stringy Floppy, then? In regard to data storage, it is approximately 13.5 times faster than cassette. A five foot wafer takes only about nine seconds to cycle in this program. Each five foot wafer will store 20 or 30 files, using the high speed modification (described by Dennis Kitsz, 80 Microcomputing, February, p. 50.) Stored on cassette, the same data takes approximately 121 seconds to cycle.

Four Commands

This program has four main commands represented by numbers. I have used INKEY$ to eliminate the ENTER key wherever possible.

When started, lines 10-50 initialize the program and make sure the data wafer is in the ESF. Line 50 assumes that you have protected your software wafer by removing the silver dot. Sample 1 shows this display. When the data wafer is inserted, write enabled, port 240 will contain a 114 and the program will continue. Lines 60-130 react according to whether your wafer has data on it. All program instructions preceded by an "@" are the Stringy Floppy instructions.

Lines 140-210 produce the function list display, tell you how many files are on your wafer (lines 940 & 950), and direct the program to the proper routine. Sample 2 shows the function list display. The commands include Command One, "FIND FILE." This command allows you to locate a particular file, based on the first and last names.

```
REMOVE SOFTWARE WAFER
& INSERT DATA WAFER
```

Sample 1. Initial display.

Command Two, ADD NEW FILE, allows you to add a file, right? These files need not be entered alphabetically, and there is no sort routine in this program since access time, using the Stringy, is relatively fast on program initialization. Searching the arrays takes less than one second with a five-foot wafer (20 records).

```
FUNCTIONS
1 = FIND FILE
2 = ADD NEW FILE
3 = CHANGE FILE
4 = END PROGRAM
THIS DATA WAFER CONTAINS 0 FILES.
PRESS THE NUMBER
OF YOUR CHOICE.
```

Sample 2. Function List.

Command Three, CHANGE FILE, allows you to change the contents of a file. CHANGE FILE has its own set of six commands, which I will describe later. Sample 3 shows this display.

```
CHANGE FUNCTIONS
1 = FIRST NAME(S)
2 = LAST NAME.
3 = STREET ADDRESS.
4 = CITY, STATE, & ZIP CODE
5 = PHONE NUMBER.
6 = FILE THE CHANGE.
PRESS THE NUMBER
OF YOUR CHOICE.
```

Sample 3.

Find File

FIND FILE is located between lines 240 and 340. The program asks what the first and last names are. Upon entering this, a FOR-NEXT loop is initiated to search the arrays. The upper limit of this loop is set by the variable F which is stored on wafer and updated when a file is added. F is initially set in line 70 or 90.

As each file is searched at line 290, line 300 uses MID$ to break down the length of the names in the file for comparison. This eliminates the need to enter the full name. What you enter must match the first set of letters. If none is found, line 310 is brought into play, and you are given another chance (line 330). Also, if you have just started the program with a new wafer, line 260 detects it and lines 270 and 280 are displayed.

When the file sought is found, it is printed via the subroutine at line 960. Line 330 allows you to search for another file, or return to the function list.

Add New File

Command Two is contained between lines 350 and 480. It is straightforward in its function and allows you to establish a new file using the INPUTs in lines 390-430. These files don't have to be name, address, etc. You can change these to whatever you wish.

Upon completing your entries, the file is displayed so you can check if it is correct. You are also told how many files the wafer now contains. If there is a typo, you can correct the file while it is still displayed. You need only correct the mis-typed fields and hit ENTER for the rest.

If everything is OK, and you are not adding more files, the new data is written on your wafer, and F is updated (lines 870-930).

Change File

CHANGE FILE is the third major command. CHANGE FILE is located between lines 490 and 840 and is the most complex. As in Command One you are asked for the first and last names. Then the arrays are searched, the files compared with your entries (line 520) and you are informed if the file is missing (line 530) or if it is displayed.

Now, you are asked if it is correct (line 850). The program responds by asking if you wish to change another file, or it displays the six commands which allow you to change each field of the file separately. Each displays what the field contains, asks if it's correct, and then returns to the change functions, or allows you to input the new information. Again, you are asked to see if it is correct.

Command Six rewrites the file onto wafer after asking if you wish to change another file. You are then returned to the function list.

I have used a number of subroutines between lines 850 and 1010 which saves memory and allows the program to be consistent in various displays.

Lines 1020 through 1110 are the system error trapping routines and are set in line 10. I find these routines helpful, especially when I have forgotten to load the ESF Data I/O software, or used the wrong wafer.

Note that line 350 checks if the wafer is full. I have limited the total to 80 files since a 20-foot wafer holds 8000 bytes and most files are 100 bytes in length, or less. Also, a 16K system like mine does not have enough memory to store the 200 files that a 50-foot wafer would hold.

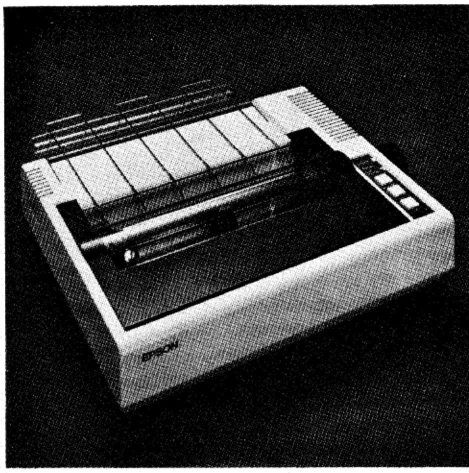
Your first thought upon reading this article, was probably that this is another data base management system. I suppose it is, but, there is no attention to alphabetizing and, no provisions for printing information on a printer, or deletion of files. This saves memory and allows the program to perform its purpose, to be a quick and easy method to store and retrieve your personal phone book.

As for file deletion, files can be changed to a first name of SPARE and a last name of FILE and the other three fields changed to a single space. ■

Program Listing

```
10 CLEAR8500:ONERRORGOTO1020
20 DIMNF$(80),LN$(80),SA$(80),CS$(80),PN$(80)
30 CLS:PRINTTAB(22);"STRINGY PHONE BOOK":PRINT:PRINT:PR
   INT
40 PRINTTAB(10);"Remove software wafer & insert data wa
   fer."
50 IFINP(240)<>114THEN50
60 CLS:PRINT"Is this a new data wafer (y/n)?":GOSUB860
70 IFI$="y"THENF=0:GOTO140   :ELSEIFI$="n"THEN80   ELSE6
   0
80 @OPEN1
90 @INPUTF
100 FORSF=1TOF
110 @INPUTNF$(SF),LN$(SF),SA$(SF),CS$(SF),PN$(SF)
120 NEXT
130 @CLOSE
140 CLS:PRINTTAB(22);"STRINGY PHONE BOOK":PRINT
150 PRINTTAB(26);"Functions":PRINT
160 PRINTTAB(22);"1 = Find file."
170 PRINTTAB(22);"2 = Add new file."
180 PRINTTAB(22);"3 = Change file."
190 PRINTTAB(22);"4 = End program.":PRINT:GOSUB940   :GO
    SUB1010
200 GOSUB860   :IFVAL(I$)<1ORVAL(I$)>4THEN140
210 ONVAL(I$)GOTO240   ,350   ,490   ,220
220 CLS
230 PRINT"*** END OF RUN ***":END
240 CLS:INPUT"The first name(s) you are looking for is
    (are)";SF$
250 INPUT"And the last name";SL$
260 PRINT:IFF>0THEN290
270 PRINT"There is no data on file."
280 PRINT"Hit 'ENTER' for the Function List.":GOSUB860
    :GOTO140
290 FORSF=1TOF
300 IFSL$=MID$(LN$(SF),1,LEN(SL$))ANDSF$=MID$(NF$(SF),1
    ,LEN(SF$))THEN320   ELSENEXT
310 PRINT"File not found.":PRINT:GOTO330
320 GOSUB960
330 PRINT"Do you wish to search for another file (y/n)?
    ":GOSUB860
340 IFI$="y"THEN240   ELSEIFI$="n"THEN140   ELSE330
350 CLS:IFF=80THENPRINT"Your data wafer is full.":ELSE3
    80
360 PRINT"Insert a new data wafer."
370 IFINP(240)>=115THEN370   ELSE50
380 F=F+1:CLS
390 INPUT"The first name(s) is (are)";NF$(F)
400 INPUT"The last name is";LN$(F)
410 INPUT"The street address is";SA$(F)
420 INPUT"The city, state, & zip code is (no commas)";C
    S$(F)
430 INPUT"The phone number is";PN$(F)
440 SF=F:GOSUB960   :GOSUB950   :GOSUB850
450 IFI$="y"THEN460   ELSEIFI$="n"THEN390   ELSE440
460 PRINT"Do you wish to add another file (y/n)?":GOSUB
    860
470 IFI$="y"THEN350   ELSEIFI$="n"THEN480   ELSE460
480 GOSUB870   :GOTO140
490 CLS:INPUT"The first name(s) you wish to change is (
    are)";SF$
500 INPUT"And the last name";SL$
510 PRINT:FORSF=1TOF
520 IFSL$=MID$(LN$(SF),1,LEN(SL$))ANDSF$=MID$(NF$(SF),1
    ,LEN(SF$))THEN540   ELSENEXT
530 PRINT"File not found.":GOTO560
540 GOSUB960   :GOSUB850
550 IFI$="y"THEN560   ELSEIFI$="n"THEN580   ELSE540
560 PRINT"Do you wish to change another file (y/n)?":GO
    SUB860
570 IFI$="y"THEN490   ELSEIFI$="n"THEN140   ELSE560
580 CLS:PRINTTAB(23);"CHANGE FUNCTIONS":PRINT
590 PRINTTAB(21);"1 = First name(s)."
600 PRINTTAB(21);"2 = Last name."
610 PRINTTAB(21);"3 = Street address."
620 PRINTTAB(21);"4 = City, state, & zip code."
630 PRINTTAB(21);"5 = Phone number."
640 PRINTTAB(21);"6 = File the change.":PRINT
650 GOSUB1010 :GOSUB860   :IFVAL(I$)<1ORVAL(I$)>6THEN580

660 ONVAL(I$)GOTO670   ,700   ,730   ,760   ,790   ,820
670 CLS:PRINT"The first name(s) is (are):  ";NF$(SF):PR
    INT
680 GOSUB850   :IFI$="y"THEN580   ELSEIFI$="n"THEN690   EL
    SE680
690 INPUT"The first name(s) should read";NF$(SF):GOTO67
    0
700 CLS:PRINT"The last name is:  ";LN$(SF):PRINT
710 GOSUB850   :IFI$="y"THEN580   ELSEIFI$="n"THEN720   EL
    SE710
720 INPUT"The last name should read";LN$(SF):GOTO700
730 CLS:PRINT"The street address reads:  ";SA$(SF):PRIN
    T
740 GOSUB850   :IFI$="y"THEN580   ELSEIFI$="n"THEN750   EL
    SE740
750 INPUT"The street address should read";SA$(SF):GOTO7
    30
760 CLS:PRINT"The city, state, & zip code reads:  ";CS$
    (SF):PRINT
770 GOSUB850   :IFI$="y"THEN580   ELSEIFI$="n"THEN780   EL
```



```
   SE770
780 INPUT"The city, state, & zip code should read (no c
    ommas)";CS$(SF):GOTO760
790 CLS:PRINT"The phone number reads:  ";PN$(SF):PRINT
800 GOSUB850  :IFI$="y"THEN580  ELSEIFI$="n"THEN810  EL
    SE800
810 INPUT"The phone number should read";PN$(SF):GOTO790
820 CLS:PRINT"Change filed.":PRINT"Do you wish to chang
    e another file (y/n)?"
830 GOSUB860  :IFI$="y"THEN490  ELSEIFI$="n"THEN840  EL
    SE820
840 GOSUB870  :GOTO140
850 PRINT"Is this correct (y/n)?"
860 I$=INKEY$:IFI$=""THEN860  ELSERETURN
870 @OPEN1
880 @PRINTF
890 FORSF=1TOF
900 @PRINTNF$(SF),LN$(SF),SA$(SF),CS$(SF),PN$(SF)
910 NEXT
920 @CLOSE
930 RETURN
940 PRINTTAB(15);
950 PRINT"This data wafer contains";F;"files.":PRINT:RE
    TURN
960 CLS:PRINT"The file reads:":PRINT
970 PRINTTAB(5);NF$(SF);" ";LN$(SF)
980 PRINTTAB(5);SA$(SF)
990 PRINTTAB(5);CS$(SF)
1000 PRINTTAB(5);PN$(SF):PRINT:RETURN
1010 PRINTTAB(15);"Press the number of your choice.":RE
     TURN
1020 IFERR/2+1=2THEN1050 ELSEIFERR/2+1=4ORERR/2+1=22THE
     N1080
1030 CLS:PRINT"You have made a mistake.":PRINT"Please t
     ry again."
1040 GOTO1100
1050 CLS:PRINT"You have forgotten to load your ESF Data
     I/O."
1060 PRINT"You will have to start over after first load
     ing the Data I/O."
1070 RESUME230
1080 CLS:PRINT"You have used the wrong data wafer."
1090 PRINT"Insert the correct wafer and try again."
1100 @CLEAR:PRINT:PRINT"Hit 'ENTER' when ready.":GOSUB8
     60
1110 RESUME60
```

Don't face these things like a cowardly lion!.

Compile, Interpret, Assemble. . .OH BOY

Ed Faulk
2531 E. Commonwealth Ave.
Fullerton, CA 92631

Whenever a newcomer to computing hears the terms compilers, assemblers, or interpreters, I am reminded of the look on the face of the Scarecrow in "The Wizard of OZ" when Dorothy says, "Lions and tigers and bears." Stark terror crosses the face followed by shortness of breath and cold sweats. The concepts are actually relatively simple to grasp once you understand how a computer executes a program.

To Begin

Let's start with the interpreter. This ubiquitous piece of code is usually the first encountered by the novice home computerist.

If you've ever seen a foreign film with English subtitles or witnessed a session at the United Nations, you've seen the human communication equivalent of an interpreter. The interpreter provides the meaning or sense behind the actual spoken expressions. In a computer, the BASIC interpreter performs the same function.

The interpreter scans a line of BASIC code, determines the "verb," "subject" and "object," and then conveys the meaning to the computer. The process of scanning the code is called parsing, and the extracted parts of the code are the command (verb) and any operands that might be present (object and subject). Once the discrete items have been isolated and their addresses located, the interpreter calls a routine based on the command which will effect the desired action.

This process is repeated for all of the instructions that are executed in a program. If a given line in the program is executed multiple times, it is interpreted multiple times.

A compiler is more akin to a translator than anything else in human experience. This software product takes a program in a language such as BASIC, COBOL or FORTRAN (called a source program) and translates it into machine language so the computer can act on the resultant program (called the object program) directly. It should be mentioned that the object program is usually processed through a program called a linkage editor. This program brings all of the modules used in the program into a single executable module which is then called a load module.

The function of determining what code should be generated for the various source statement lines is handled by the compiler in much the same fashion as the interpreter. The line is parsed and the components determined. The appropriate code is then generated based on the types of data involved and the operation to be performed.

Multiple Passes

Most compilers make multiple passes through a program to perform the various activities that must be performed. Such passes can include the identification of the various data items and their types (numbers, both binary and BCD, and character strings). After identifying these items they have addresses assigned and their attributes (types) noted so that later passes will generate the correct code when the data items are acted upon.

Almost without exception it can be stated that where there is a computer there is an assembler. An assembler is the bits and bytes method of preparing a program for execution. As you get further away from the basic elements of any operation the capability to manipulate discrete events is more and more restricted. In BASIC there are some concessions to bit manipulation through the use of commands such as PEEK and POKE and the primitive Boolean functions of AND, OR, and NOT.

Other high level languages normally do not have such facilities, although Microsoft FORTRAN does. Even with these functions, the testing or setting of a single bit of data becomes extremely difficult. In machine (assembly) language it is usually very simple. The Z-80, for example, supports the BIT command for testing a single bit, the RES command for resetting a single bit, and the SET command for setting a bit. Masking or selecting data is supported with commands like AND, OR and XOR (exclusive OR). Ones and zeros can be reversed (the complement) or negated (reverse the sign). These functions can be simulated in the higher level language at the expense of execution time.

The assembler takes machine language and converts it on an instruction-for-instruction basis into code that is directly executable. The primary difference is that a compiler will generate many instructions at the machine level for a single high level instruction. This is part of the reason that machine language programs can often run circles around compiled programs.

Assembly language is harder to learn than any high level language because the programmer must take the idiosyncrasies of the computer into consideration, while the high level language programmer is free to consider only the problem to be solved. It is for this reason that high level compiled languages are often called problem oriented languages.

Why would one select a compiler over an interpreter or an interpreter over an assembler? The answer lies in the experience of the user and the result desired. There are several considerations involved in making this choice, including desired speed of execution, development time available, experience of the programmer, etc.

The obvious advantage of an interpreter is the ability to debug the program easily. Normally one codes a program and then starts testing it. The BREAK key usually provides a way to halt execution of the program so that the values stored in the various data areas can be examined and modified as needed. The major disadvantage of this system is slow execution speed.

Using an assembler is far more tedious for development since there is much more code that needs to be written, and thus there is a greater chance for errors to creep into the code undetected. Debugging, while not as easy as for an interpreted language, is not too difficult with the aid of a monitor program that allows single stepping (executing one instruction at a time) through the program and will display memory on command. Also available are the locations of the various routines. The monitor allows the programmer to change data or code to patch the program until a final fix can be made.

The final candidate, the compiler, is still more difficult to debug. Coding is as simple as with the interpreter since this is a higher level language, but debugging is far more difficult. Part of the difficulty lies in the problem of determining the code that was actually generated for each statement. Often the compiler will have an option that allows this, but sometimes it omits the locations of the data. The generated code is often not as efficient as it could have been had a programmer written it instead of a compiler. As with the assembler, a monitor is necessary for debugging.

With both the assembler and the compiler, the program must be re-edited and the changes made. The new program must be recompiled, and re-link edited before execution. The interpreter simply allows editing to correct the error or omission, and re-running the program to continue testing.

The obvious compromise is to develop a program using an interpreter, and then compile the debugged version. This allows the best of both worlds. The interpreter uses the debugging for development, and the compiler's speed delivers the final product. The assembler should be used for those that don't have a compiler and need greater execution speed.

The choice of development tools is up to each programmer, but my own preference has been to model (or simulate) programs in BASIC until I am sure they work right, and then to convert to FORTRAN if I want speed. I use the assembler for those occasions when I want either super speed, or when FORTRAN isn't usable. ■

✓Reader Service—see page 242

Let your computer do the teaching.

Vocabulary Builder

Roger Zimmerman
9614 NE Hazel Dell Ave.
Vancouver, WA 98665

The following program is designed primarily to help you build your knowledge of a foreign vocabulary, but it has other uses, too. It can teach you the vocabulary of a specific discipline, like biology or physics. It can even increase your English vocabulary.

Building Vocabulary

The first time you use this vocabulary program you will need to answer the keyboard-tape question with k, for keyboard. It will then ask you to enter a word and its definition (use *** for word to stop). You may enter foreign words and English definitions or English words and foreign definitions, or mix them. The word will be the unknown and the definition will be the clue.

Allow the computer to quiz you until you are competent with your vocabulary list. It is helpful to pronounce the words as you type them, to associate the pronunciation with the meaning.

Look at the chart to see if there are any words that are stumbling blocks. You may wish to make a special tape of these difficult words (the computer will ask you whether you do). You may want to make a tape of the entire list (answer N to the difficult words question, and Y to the complete words question).

Each time you return to the program, add two or three words to the list.

After each session, if you add any new words, the computer will ask if you wish to save the list again. I have had no problem taping the new list right over the old list by starting the tape in about the same place (my standard procedure is to start all computer tapes at counter 005).

When the list gets long, start a new one or weed out all of the easy words on the old list. I prefer to save my old lists for periodic review.

By the way, you have 10000 available string space bytes and a 100-item list limit.

Note the use of CHR$(27) in lines 2015 and 2035. This ASCII code scrolls the cursor up one line. By using this just before the INPUTed data is printed on the screen, the new data will print where the prompt was. Then CHR$(30) must be used to clean off the rest of the prompt. The use of CHR$(8) here is to remove the space between the entry number and the period from PRINT I:;".".

Now look at line 1040. Note that the definition is printed on the screen first at TAB(23) plus one space " ". The cursor is then returned to the beginning of the line and the word is printed. Sometimes the word is just long enough to push the left margin of the definition into the next field on the screen, but not long enough to cover it if it were to be left in the second field (assuming you use PRINT W$(I),D$(I)). By doing it as in line 1040, the left margin of the definition is always even, except when the word is really long. Then, the definition will begin just one space past the word.

You may also wonder about line 131. This allows the INKEY$ input to print in the right place.

To have the computer turn on

continues to p. 228

Program Listing

```
1 CLEAR10000:DIMW$(100),D$(100),W(100),T(100)
10 CLS:I=1
20 PRINT"FOREIGN LANGUAGE VOCABULARY PRACTICE"
30 FORX=0TO1000:NEXT
40 CLS
50 PRINT"DO YOU WISH TO LOAD VOCABULARY BY TAPE OR BY K
     EYBOARD";
60 A$=INKEY$:IFA$=""GOTO60
70 IFA$<>"T"ANDA$<>"K"GOTO40
80 IFA$="T"GOSUB1000
90 IFA$="K"GOSUB2000
99 II=I
100 CLS:PRINT"I WILL NOW GIVE YOU SOME DEFINITIONS AND
     ASK YOU TO GIVE ME THE CORRESPONDING VOCABULARY WO
     RD. THE WORD MUST BE EXACT SPELLING  AND MUST BE E
     XACT LEXICAL FORM."
105 PRINT:INPUT"PRESS ENTER TO CONTINUE";A
106 CLS
110 RANDOM:I=RND(II)
120 PRINT@0,CHR$(30):PRINT@0,D$(I)
125 PRINTCHR$(30);:PRINT"WHAT IS THE WORD?";:PRINTCHR$(
     30)
130 PRINT@896,"TO END TYPE 'E'"
131 PRINT@128,"":T$="":P=0
132 S$=INKEY$:P=P+1:IFP=250PRINT@512,CHR$(30):PRINT@768
     ,CHR$(30):GOTO132ELSEIFS$=""GOTO132
133 IFASC(S$)<32ANDASC(S$)<>8GOTO136ELSEU$="":IFASC(S$)
     =8GOTO138ELSEGOTO134
134 T$=T$+S$
135 PRINT@256,CHR$(30):PRINT@256,T$:GOTO132
136 PRINT@256,CHR$(30):IFT$="E"GOTO190
137 U$="":IFASC(S$)=8GOTO138ELSEGOTO140
138 PRINTCHR$(8):IFLEN(T$)=1T$="":GOTO135ELSEPRINT@256,
     CHR$(30):TT=LEN(T$):FORX=1TOTT-1:U$=U$+MID$(T$,X,1
     ):NEXT:T$=U$:GOTO135
140 IFT$=W$(I)GOSUB3000
150 IFT$<>W$(I)GOSUB4000
160 GOTO110
190 GOSUB7000
200 CLS:IFA$="K"ORA$="Y"ORC=1PRINT"WOULD YOU LIKE TO SA
     VE THE COMPLETE LIST OF WORDS YOU WERE DRILLED ON
     PLUS YOUR ADDITIONS ON TAPE";
210 B$=INKEY$:IFB$=""GOTO210
220 IFB$<>"Y"ANDB$<>"N"GOTO200
```

```
230 IFB$="Y"GOSUB6000
999 CLS:PRINT"BYE FOR NOW FROM THE VOCABULARY TESTER":F
    ORX=0TO1000:NEXT:CLS:CLEAR50:END
1000 CLS:INPUT"LOAD TAPE (PLAY) AND PRESS ENTER";A
1005 CLS:PRINT"LOADING VOCABULARY . . .PLEASE STAND BY"
1020 INPUT#-1,W$(I),D$(I)
1028 IFW$(I)<>"***"GOTO1040
1030 IFW$(I)="***"PRINT"WOULD YOU LIKE TO ADD TO THIS L
     IST?"
1031 A$=INKEY$:IFA$=""GOTO1031
1032 IFA$="N"I=I-1:RETURN
1033 IFA$="Y"C=1:GOTO50
1034 IFA$<>"Y"ANDA$<>"N"GOTO1031
1040 PRINTTAB(23)" "D$(I)CHR$(29)ICHR$(8)". "W$(I):I=I+
     1
1050 GOTO1020
2000 CLS
2010 INPUT"ENTER WORD ('***' TO END)";W$(I)
2015 PRINTCHR$(27)CHR$(30)ICHR$(8)". "W$(I)
2020 IFW$(I)="***"I=I-1:RETURN
2030 INPUT"ENTER DEFINITION";D$(I)
2033 IFPEEK(15384)=32PRINT
2035 PRINTCHR$(27)CHR$(27)CHR$(27)ICHR$(8)". "W$(I)TAB(
     23)" "D$(I)CHR$(30)
2040 I=I+1
2050 PRINT:GOTO2010
3000 R=R+1
3002 PRINT@768,CHR$(30)
3005 T=T+1
3007 T(I)=T(I)+1
3010 RESTORE:Q=RND(10)
3020 FORX=1TOQ:READQ$:NEXT
3030 PRINT@512,Q$;:PRINTCHR$(30)::PRINT@640,"THAT'S";R;
     "CORRECT OUT OF";T;"OR";(R*100)/T;"%";:PRINTCHR$(3
     0)
3040 IFT$<>W$(I)PRINT@768,D$(I);" = ";W$(I);:PRINTCHR$(
     30)
3050 RETURN
4000 W=W+1
4005 W(I)=W(I)+1
4010 T=T+1
4015 T(I)=T(I)+1
4020 RESTORE:Q=10+RND(10)
4030 GOTO3020
5000 DATA"RIGHT","CORRECT","VERY GOOD","YOU GOT IT","GO
     OD","ABSOLUTELY RIGHT","THAT'S IT!","WHADDYA KNOW
     THAT'S RIGHT","NICE WORK - THAT'S IT","GOOD GOIN'
     CHARLIE, YOU'RE RIGHT","WRONG","INCORRECT","OOPS",
     "SORRY","STUDY THAT ONE SOME MORE","YOU GOOFED"
5010 DATA"NOP","BITE YOUR TONGUE!","YOU SHOULD KNOW BET
     TER","DO YOU WANT TO KICK YOURSELF NOW?"
5999 END
6000 CLS:INPUT"INSERT TAPE (RECORD) AND PRESS ENTER";Q
6001 CLS:PRINT"WRITING ONTO TAPE NOW"
6010 FORI=1TOII:PRINT#-1,W$(I),D$(I)
6012 PRINTICHR$(8)". "W$(I),TAB(24)D$(I):
6015 NEXT
6020 PRINT#-1,"***","***"
6030 RETURN
7000 CLS:PRINT"NOW I WILL SHOW YOU WHICH WORDS NEED THE
      MOST WORK."
7010 PRINT:PRINT"FOLLOWING IS A LIST OF EACH WORD YOU M
     ISSED AT LEAST ONCE       FOLLOWED BY THE NUMBER O
     F TIMES YOU MISSED IT AND OUT OF HOW     MANY TRIES
     ."
7020 PRINT:INPUT"PRESS ENTER TO CONTINUE";A
7030 GOSUB9000
7040 PRINT:FORI=1TOII
7050 IFW(I)>0PRINTW$(I),W(I),T(I),((T(I)-W(I))*100)/T(I
     );"%"
7070 B=PEEK(16305):IFB<>32ANDB+1<>32PRINT"PRESS ENTER T
     O CONTINUE";:INPUTA:GOSUB9000
7080 NEXT
7082 INPUT"* END OF LIST *          PRESS ENTER TO CONTINU
     E";X
7092 PRINT:PRINT"NOW IF YOU WUOLD LIKE TO SAVE ANY OR A
     LL OF THESE 'DIFFICULT'    WORDS ON TAPE FOR LATER
     'SPECIAL' REVIEW, YOU WILL THEN BE GIVENA CHOICE O
     F THE LEVEL OF DIFFICULTY OF THOSE WORDS YOU WISH
     TO  SAVE."
7093 PRINT:PRINT"WOULD YOU LIKE TO SAVE ANY? (PRESS 'R'
      TO REVIEW LIST)"
7094 S$=INKEY$:IFS$=""GOTO7094
7095 IFS$="R"GOTO7030
7096 IFS$="N"GOTO200
7097 IFS$<>"Y"ANDS$<>"N"ANDS$<>"R"GOTO7094
7098 IFS$="Y"GOSUB10000
8000 RETURN
9000 CLS:PRINT"WORD","TIMES MISSED","TIMES TRIED","% TI
     MES RIGHT":RETURN
10000 PRINT:PRINT"YOU WOULD LIKE TO SAVE ANY WORDS YOU
     ONLY GOT RIGHT --- % OF THETIME OR LESS. FILL IN T
     HE BLANK.";:INPUTQ
10010 PRINT:INPUT"PREPARE TAPE (RECORD) AND PRESS ENTER
     ";X
10015 CLS:PRINT"WRITING TAPE NOW"
10020 FORI=1TOII:IFT(I)>0IF((T(I)-W(I))*100)/T(I)<=QPRI
     NTW$(I),D$(I):PRINT#-1,W$(I),D$(I)
10025 NEXT
10027 PRINT#-1,"***","***"
10030 RETURN
```

and off messages while waiting for an INKEY$ input, look at line 132. Usually the INKEY$ function works in a sort of scanning loop (i.e., 10 A$ = INKEY$:IFA$ = ""GOTO10). This way, if no input is encountered it starts searching again. Suppose in-between each scan we have the computer add one to a variable. Before it goes back for the next scan, it can increase the variable, and after the variable reaches a certain point, it can do whatever we want (in this case, turn off two of the lines on the screen). Be careful not to cram too much in or the computer might miss the input while doing the other job.

How about getting the INKEY$ function to accept any length word? Look at line 133. Anything that is not a printable character, but is less than ASC$(32), will trigger a break from the search-for-another-letter routine. The ENTER key functions as it normally does with an INPUT statement. Otherwise the computer would look back and keep searching for another letter to add to the current string. Why didn't I just use an INPUT statement? Because I couldn't have turned off those lines while searching for an input.

One exception is that if the computer sees a CHR$(8), it goes to a special routine (line 138), the backspace arrow. This line counts the number of letters in the present input string and subtracts the last one. This doesn't work if there is just one character. The word would then become a "" (null string) and you are ready to start the answer again.

Now the toughies. On that list of difficult words, how do I keep from scrolling right off the screen when it is full? It sounds like a job for Super Machine Routine. Not so. If Radio Shack could do it in BASIC in their Budget Management program, I can do it in my Vocabulary Builder program. Radio Shack counts the number of lines it prints, then goes to press-enter-to-continue.

I did it this way. I remembered that the screen positions are treated as part of memory, and that PEEK could look and see what ASCII code was in any position of memory. If I PEEK at a spot where the bottom of the readout always changes the screen, and test it to see if it is still blank, I know whether the screen is full. Look at line 7070.

Position 16305 is the spot on the screen where the first letter of the fourth column (field) of the last line I wanted is located. If PEEK(16305) ever becomes anything besides a blank space (CHR$(32)), then I know the screen is full and I go to press-enter-to-continue.

A cassette is available of this program from the author. ■

MAKE $$$$$

$$$$ MONEY

Selling 80 Microcomputing, the only major journal for the users of the TRS-80*, is a sure bet for getting the computer enthusiast into your store. Once through the door you can sell him anything.

We know "80" will make you money . . . it's the only magazine for the TRS-80* users and you know how many of those there are. So call today and join the dealers who make money with "80".

For information on selling 80 Microcomputing, call 603-924-7296 and speak with Ginnie Boudrieau, our Bulk Sales Manager. Or write to her at 80 Microcomputing, Pine Street, Peterborough, NH 03458.

***TRS-80 is a trademark of the Tandy Corp.**

VISA/MC Order Line Only (except Mich.) 800-253-4358 ext. 100

FINDISK-II The ultimate in automatic disk indexing with exclusive features ..... Model-I $20.00
AUTOMATICALLY create INDEX of programs or data from all your disks, print disk LABELS, print alphabetized MASTERLIST, do fast SEARCH, add DESCRIPTIONS. Also automatically: detect DATA or SYS disks, PURGE disks and index of old files, and UPDATE from revised disks only.

SOLAR-I The critical calculations for passive design. . . Model-I $30.00 .... Model-II $45.00
INPUT: any latitude, orientation, slope, roof overhang, storage type, building loss, OUTPUT: solar angles, shading, time, heat gain/loss, percent solar, fuel use. Print report by hour, month, year in presentation format.

RIA-II Complex Real Estate Investment Analysis .... Model-I $30.00......Model-II $45.00
Analysis for investor or homeowner using Elwood method. INPUT: Project costs, loan and tax data, expenses, depreciation rate. OUTPUT: Cap rate/value, mortgage payments, before/after tax cash flows, return (IRR), profit/gain from sale over any time series.

DEPRECIATE-I Manage a list of depreciable items .... Model-I $15.00.....Model-II $20.00
Tracks long list of depreciable items with varying purchase dates, depreciation rates, or per cent business use. Update any time. Print tax form. Used by many CPAs.

STRUCT-I Graphic design of steel/wood beams and moment transfer............ Model-I $15.00
INPUT: span/cantilever, uniform/point loads, beam material. OUTPUT: with screen graphics beam moment & shear diagrams. Print job report with diagram, stresses, and required beam sizes.

Min 32K. On disk (Mod-I one drive order tape). Add $1.00 postage (Mich. add 4% tax) VISA/MC

✓88 DOCUMAN SOFTWARE BOX 387-A KALAMAZOO, MI 49005 (616) 344-0805

By Gregory Berrchill

The Magic Cursor allows you to easily create screens (including graphics) on your video. A powerful command then generates the BASIC instructions to recreate the screen.

The Magic Cursor makes sophisticated Data Entry and Display easy. With **the Magic Cursor**, you define the data entry or display fields directly on your screen. The definition commands will generate the BASIC instructions to implement data entry and display.

The Magic Cursor will also
- create graphical characters simply on the screen
- duplicate characters on the screen easily
- store screens for later use in the same session
- reload screens created in previous sessions
- center blocks of text on the screen
- move blocks of text on the screen
- right justify text within a block on the screen

and more . . . it has 27 powerful commands!!

MAGIC CURSOR (32K Disk) Mod I **$99.95**

Trace-80™

TRACE-80 lets you observe the inner working of a machine language program. It allows you to run a machine language program in slow motion and watch the screen. You can stop execution at any time and examine the contents. You can execute your program and watch each instruction mnemonic and register contents list to the screen in place of normal screen display.

If you have a printer, **TRACE-80** allows you to execute your program in slow motion and watch the screen while your printer simultaneously prints the machine code being executed, the memory location and the instruction mnemonic along with the current register contents.

TRACE-80 allows you to trace ROM as well as RAM because instructions are emulated in a special execution buffer.

TRACE-80/Mod I (16K Tape or Disk) **$29.95**
TRACE-80/Mod II **$49.95**

The Restauranteur's Consultant™

This food and beverage management tool dramatically reduces the human factors in food cost analysis. Those tasks required to effectively operate any restaurant or food service business. Over a half dozen reports give uniform, accurate and up to the minute information for profitability. And handling of daily cost changes requires only minutes per week, instead of hours because the Consultant makes all the necessary conversions from your case prices.

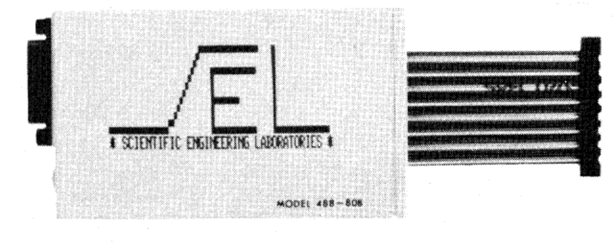
Mod I, 32K, 2 Disk, Printer or Mod II **$750.00**

CUSTOM COMPUTER CENTER, INC.

For Ordering or Information Write
P.O. Box 58042 Houston, TX 77058
Attn: Jim Martens
or Call (713) 474-2428

✓7

WRITE FOR OUR COMPLETE SOFTWARE CATALOG!!



IEEE-488 to TRS-80* INTERFACE

Mod. 488-80B

$225.00
+ shipping, insurance & tax

SPECIFY DISK OR TAPE

For Model 3 Operation Contact Factory

Everything needed to add *powerful GPIB-488 controller capability* to TRS-80, Model 1, Level 2 or DOS

SCIENTIFIC ENGINEERING LABORATORIES
11 NEIL DRIVE • OLD BETHPAGE, NEW YORK 11804
TELEPHONE (516) 694-3205 ✓291

*Trade Mark of Tandy Corp.
There is no affiliation between Scientific Engineering Laboratories and Tandy Corporation or Radio Shack.

✓ Reader Service—see page 242

A real hands-on report on the Shack's TV, RF modulator board.

Perverse Video

William P. Winter, Jr.
O'Higgins 3168
1429 Buenos Aires
Argentina

I wanted to purchase a TRS-80, but I had to figure the best way to get it up and running at the lowest initial cost. I had space problems too. I was traveling by air to Argentina with my suitcases already bulging.

I could purchase the CPU (central processing unit) for $400. This included the keyboard, but no peripherals. I could use my own cassette recorder and TV, already in Argentina. I tried to find an RF modulator, but the only one I located put out a UHF signal. My TV set had VHF only.

Then a dealer told me that Radio Shack would have a VHF interface on the market shortly. I got one just before I left.

RF Modulator

It comes as a project board with printed circuit board, modulator subassembly, antenna switch box, front panel label, and instruction book. All other parts are stock items and can be purchased from Radio Shack, or you can use parts from your junk box. The manual is very complete, the instructions are clear and the kit goes together easily.

My brother offered me a burned out 12-inch G.E. portable. A trip to a local Radio Shack store to test the tubes showed one with a burned out filament. I replaced the tube, and a channel select knob; did some additional filtering in the power supply, and the old TV worked fine.

The interface can be powered from most sources. I opted to power it from the TRS-80. On page 7 the manual calls for the TRS-80 video out cable to be soldered to pin 1, which is the interface RF out connection. This is obviously a misprint, and has probably been corrected by now. I followed what seemed to be the correct connection, soldering the TRS-80 video out cable to the video input terminals of the TV interface board.

The hook-up and applications section of the manual suggests that the RF output cable be wrapped into a ten-turn coil one and one-half inches in diameter. I found that my cable was too short to get ten turns around a toilet tissue roll form. A smaller diameter form works better and is neater. I found that six turns were the minimum needed to eliminate the hash generated by the high speed switching integrated circuits in the CPU. This hash is transmitted along the shield of the cable going to the antenna input of the TV. The coil forms a choke to block this RF hash.

Isolation Transformer

Make sure that the TV is transformer-operated: If it is not, use an isolation transformer to protect the interface and the TRS-80 from dangerous line voltages. I had an isolation transformer in the junk box, but two filament transformers connected back to back would do the job.

To use, first make sure the filament windings are capable of handling the power required for the TV. For a typical small TV requiring 110 watts such as mine, the following transformer ratings are adequate.

6 V @ 20 A
12 V @ 10 A
17 V @ 7 A
24 V @ 5 A
36 V @ 3.3 A

A possible source of transformers are old TVs. Some of the older sets had 20 tubes and 2 or 3 filament windings. The newer sets have fewer tubes and therefore not enough current rating for this job. Be sure to tape off any unused windings.

When connecting windings in series measure the voltage. If two windings are out of phase when connected, the voltage will cancel. If this happens, reverse the connections of one of the windings.

Color codes for the windings of power transformers are shown in Table 1. ■

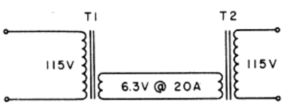


Fig. 1. Filament Transformers

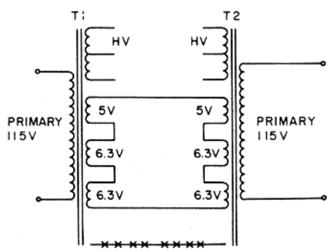


Fig. 2. Scrap TV Transformers

1. Primary leads	Black
If tapped:	
Common	Black
Tap	Black/yellow striped
Finish	Black/red striped
2. High voltage winding	Red
Center-tap	Red/yellow striped
3. Rectifier filament winding	Yellow
Center-tap	Yellow/blue striped
4. Filament winding No. 1	Green
Center-tap	Green/yellow striped
5. Filament winding No. 2	Brown
Center-tap	Brown/yellow striped
6. Filament winding No. 3	Slate
Center-tap	Slate/yellow striped

Table 1.

Prefer to roll your own?
If so, take a look at this alternative to the Shack's E.I.

LNW Expansion Board

System Expansion
LNW Research
Costa Mesa, CA
$69.95

Bob Couger W6KPS
1095 McCoy #99
Santa Maria, CA 93454

Do you all too often get an OM (out of memory) error? Are you tired of waiting for a program to load or to get a few dozen data files into memory? On the other hand, maybe your Level II, 16K system is now very, very reliable and you are afraid of the problems associated with the Radio Shack expansion interface. There is another way.

Alternative

LNW Research offers a circuit board and a manual for $69.95. It claims to have all of the features of the Radio Shack expansion chassis and then some. Also mentioned is software compatibility. They further claim a "quiet bus" design that eliminates some of the memory problems.

After receiving additional data from LNW, I carefully compared their specs with those of Radio Shack. It appeared that LNW's board would do everything that Radio Shack's does, plus the serial interface was not an accessory that connects via a poor connector, as is the Radio Shack version. The board had heavy power buses and lots of bypass capacitors.

I ordered the kit last spring and received it only 30 hours later. I spent some time comparing LNW's circuitry with that of Radio Shack. Anyone building the LNW board should purchase the expansion interface hardware from Radio Shack.

The manual has a good parts list. Actually, the parts are listed in several ways. The system allows partial construction for those who don't need all the features. For instance, if you only want additional memory, and don't need a floppy disk, the manual explicitly tells you what parts to buy. On the other hand, parts are sorted alphanumerically to make it easy to order and inventory.

The hardest item to find was the cable. Radio Shack wanted $25 (love/hate... hate this time). One supplier had both ribbon connectors with one end terminated, and also 40-pin solder tail connectors, so out with the soldering iron! This is the part of the job I really hated. I detest symmetrical connectors that can be wired and plugged in upside down. These dumb unkeyed connectors are the worst design feature of all.

Anyway, all the parts *finally* were found from three suppliers, not including Radio Shack. Many of the parts (such as bridge rectifiers, SCRs) were called out as Radio Shack "276-" type part numbers.

Assembly

At a local parts store I found a 16" × 13" × 3" chassis. I wanted lots of space for a good power supply, a modem, and an interface for my ham radio. While having these mounted on a single chassis makes a neater assembly, my real objective was to keep radio interference to a minimum.

Regarding power supplies, the manual suggests buying another Radio Shack module similar to the one that powers the keyboard. Check at your local Radio Shack to see if they have a bad one laying around. Chances are they do and will sell it to you for less than $5.00. You can go home, pry it apart and replace the fuse. A schematic is included in the construction manual.

I bought separate tranformers and mounted them internally with a power line filter that hopefully keeps voltage spikes out of the logic. Shown in Fig. 1, the power supply will run four expansion interfaces.

First I mounted the board, the power transformers, and cut the slots for the ribbon connectors. Next I assembled the board per the guidelines of the manual. I used sockets for all chips. The

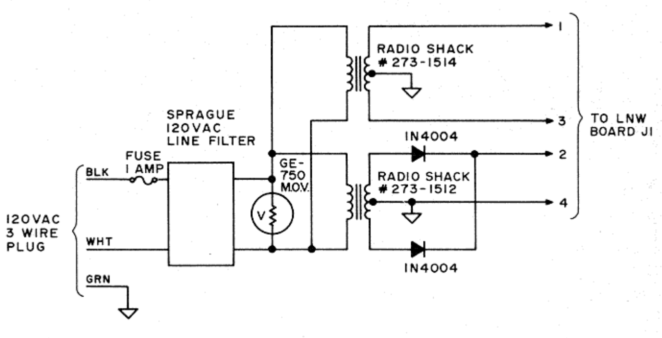


Fig. 1. This circuit will take the place of the Radio Shack power supply. The power line filter generate transients of sufficient magnitude to activate the crowbar circuits if the GE-750 is not installed. Even larger transformers may be purchased if you anticipate using floppy disks and a common supply.

pins on the sockets were tapered so that they barely extended through the board. This made inspection of soldering hard, but I managed by going slower than I normally would. There was nothing else tricky or unusual about assembling the board.

Power Up

While the manual directs you to plug in the chips prior to preliminary power checks, certain jumpers are not installed. With these removed, no power is applied to the chips. Follow the directions exactly. This section performs voltage checks to find errors or bad parts that could cause damage.

When power was first applied to my unit, both fuses blew. This was caused by the crowbar circuit, which is designed to short the power supply to ground if the regulator fails. Testing determined that turn-on transients generated by the line filter were to blame.

The problem was corrected by installation of a metal oxide varistor (MOV), General Electric Part No. GE-750. This component draws almost no current at 120 V ac but at higher voltages the resistance decreases dramatically so that spikes are shorted out. The MOV totally corrects the problem. The short probably would not have happened if I had used the Radio Shack power supply, but a MOV is good to use across the ac line of any sensitive equipment.

After this was fixed, all power supply voltages checked. I installed the jumpers, powered up the chassis, and rechecked the power supply voltages. I then removed power and connected to my keyboard.

You should hold the BREAK key when powering up a system that has an expansion chassis with no disk drives. If you don't, you will have a screen full of garbage. To recover, hold BREAK and hit RESET. I did this and the system came up with the MEMORY SIZE? prompt the way it normally does. I hit ENTER and after what seemed seconds the system jumped to BASIC in a normal manner. I learned later that this time delay is normal, since the TRS-80 has ROM resident routines that automatically check and size the memory. With the additional 32K, it takes longer.

I then entered PRINT MEM and the computer returned 48340, the correct value for a 48K machine. The next thing I tried was to PEEK and POKE into the new memory locations. Attempts to POKE or PEEK above 32767 (top of memory for a 16K machine) returned an OV error (overflow error).

Chasing the Wild Goose

While I thought it interesting that the system recognized the new memory, the fact that I couldn't PEEK or POKE into it convinced me that there was a hardware problem. Out came the old oscilloscope. I pulled out all the chips except those to support the new memory. One came out, socket and all. I had completely missed soldering it to the board. While this would not have caused my problem, I pulled the board and carefully inspected for missing parts and integrity of soldering. Finding no further problems, I then powered up and continued troubleshooting. After hours of extensive testing, I tried loading some machine language programs into the new memory.

Radio Shack's RENUM contains modules for 16, 32 and 48K machines. When I tried the 32 and 48K versions, they loaded and worked. I then loaded T-BUG and found I could write and read to high memory addresses with no problem. RSM has a memory test option, and this ran faultlessly. I began to suspect cockpit errors. After more troubleshooting, I got out my TRS-80 owner's manual, and found that to POKE and PEEK above 32767, you have to subtract 32767 from the desired memory address. Thus to POKE X into 32768, you POKE(32767 – 32768),X or POKE – 1,X. I tried it, and of course it worked.

System Expansion Port

There are two 40-pin edge connectors on the LNW board. Either one can be connected to the TRS-80. The other is available for devices that work direct-

Reader Service—see page 242

ly off the expansion port, such as a page printer. The bus termination resistors are installed near one of these connectors. The manual suggests that these resistors may not be necessary if you are using a device on the expansion connector that has its own termination networks. I installed the termination resistors and have experienced no problems.

Real Time Clock

The real-time clock is only of limited use without a disk operating system. Various programs such as the one on page 21 of Radio Shack's Expansion Interface Hardware Manual can be used to test the clock. The program seemed to work well but the clock gained about three minutes in ten hours. Using my frequency counter, I checked the frequency of the internal 4 MHz xtal. oscillator and found it running high.

By trial and error, I found that increasing the value of C15 from 27 to 39 pF put the oscillator right on frequency and made the clock accurate. The exact value varies with the tolerance of the crystal. If the frequency is high, the value of C15 should be increased, if low, vice-versa. Actually this whole effort is unnecessary since the clock loses time anyway during disk read and write operations. The clock has proven useful in disk operation for timing program events in BASIC programs.

". . . I call it magic because of a design job I did years ago using 15 chips or so to do the job that one chip does now."

Dual Cassette Operation

Cassette operation is slow, but dual cassettes open up a new world for data handling if you are interested in editing data files. The LNW board has the decoder and relay driver. All that is required in addition is an external relay which I mounted in my chassis. The system defaults to cassette recorder two when powered up; that is, if you power up and command CLOAD or CSAVE the system will activate cassette two. However, if system tapes are loaded, the relay will energize and load it from recorder one. When you have this feature enabled, you should specify CLOAD #-1 or CLOAD #-1, "A" etc. However, if you just say CLOAD or CSAVE "A", etc. the system will energize the last recorder selected from previous cassette operations.

Serial Port

The serial port uses a common chip called a UART (universal asynchronous receiver/transmitter). This chip is almost a magic device. You inject parallel data, and out comes serial data, or input parallel data and out comes serial data. The chip can do both jobs simultaneously even with unrelated data of different baud rates. Of course, it takes a clock or two and a few jumpers, but I call it magic because of a design job I did years ago using 15 chips or so to do the job that one chip does now.

Actually I like this serial port so well that I wish the LNW board had two of them. It would be nice since it is possible that one might want one port for a line printer and one for a modem. LNW includes this feature that Radio Shack charges over $100 for. You have to pur-

✓ Reader Service—see page 242

chase the additional parts, but the cost is less than $15.

Line Printer Port

The line printer port, like every other LNW feature, works like Radio Shack's and is designed to work with a parallel printer, using the Centronics interface. For those using a serial printer, the serial interface can drive it. With no modifications to the printer or the LNW board, you can set up the serial interface for the right baud rate and punch in a software routine that will link the LLISTs and LPRINTs to the serial interface. The driver software program is included in the manual.

A better way is to follow the instructions of Chapter 6 which route signals that would normally go out the normal parallel printer port through the serial port. This allows use of the LPRINT and LLIST commands without any special software. A switch is provided to allow the serial interface to be configured back to work with something like a modem.

A still better approach would be to build a separate parallel to serial converter which would plug into the printer port and convert to the type of serial level (usually RS-232) required by the printer. While either of the above techniques work well, I wanted to leave the serial port alone for a modem in the future. I constructed a small wirewrap card that is mounted inside and derives power from my Heath H-14 line printer. Since it plugs into the normal printer port, LLISTs and LPRINTs work the same as they do using a Radio Shack or Centronics printer.

Those of you who are handy with a wirewrap tool can receive a schematic of what I am using by sending me five dollars. The schematic also has optional circuitry to operate a Heath H-14 directly off the expansion port of an unmodified 16K Level II machine.

The Disk

I had my LNW chassis up and running for almost a month before my Pertec drive arrived. During this time I ran every test I could to ensure that it was working.

The Pertec box was ripped open and the four-drive cable quickly connected to the drive. Without pausing to read any directions, I plugged it into power and to the chassis interface. I then inserted the TRSDOS 2.3 system disk that I had purchased and turned everything on. The drive came on, the screen went blank, and up came DOS READY.

RF Interference

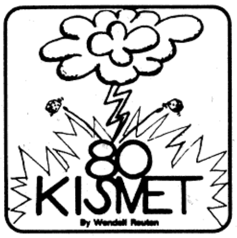
You may not care about this if you are not a radio amateur or shortwave listener. The TRS-80 generates a lot of energy in the radio frequency spectrum. It seems to be the worst in the 40 meter region. Someday I am going to hook my antenna tuner up to it and I probably will be able to talk to Japan if I can figure out some way to key it.

It seems that every accessory added causes the noise level to come up. I built my LNW board in a large metal box with the intent of building in my M80 (ham radio interface). This would eliminate one cable as a radiating source. The line filter also helps. I have done nothing to correct the radiation caused by the TRS-80 but I do not observe any increase in noise when the expansion interface is connected or disconnected.

Software Compatibility

I have tried NEWDOS+, NEWDOS 80, Electric Pencil, Percom's DOS, etc., etc. All run exactly as on a standard TRS-80. The LNW system does not experience the mysterious system crashes that some of my friends occasionally have.

The LNW board performs. It is totally hardware and software compatible with Radio Shack products designed for the TRS-80. While the manual is good, the project is such that I would recommend it only for the advanced kit builder. ■



Save and restore screen contents in 24 bytes or less.

Now You See It

Hubert C. Borrmann
2840 South Circle Drive #209
Colorado Springs, CO 80906

Recently I was working on my TRS-80, and while admiring the neat screen display, I goofed and responded with something the computer did not like.

REDO, EXTRA IGNORED and/or scrolling of the display destroyed the neat layout. I started over and decided to write a subroutine to save and restore the screen contents whenever I wished.

Since machine language is fast, I wrote the small 24-byte machine language routine (see Listing 2), which has two entry points. The first entry point saves the screen memory, 15360-16383, in upper memory, 31742-32767, and the second entry point restores screen memory with the saved memory.

Entry to the machine language is via USR(0). The following BASIC subroutine (Listing 1 and Fig. 3) does the linking and also POKEs the machine language into memory. This approach requires you to save upper memory and you should respond to MEMORY SIZE? with 31699.

Try it! You may like it. Also, see the little driver (Listing 3), which shows you how fast this all works. Maybe you do not wish to save and restore all 1024 bytes.

If your application calls for saving and restoring the upper half only, you can change the byte counter BC in Listing 2, lines 130 and 180, from 1024 to 512 and you should change 010004 to 010002 in both lines.

This means that you should change both DATA statements that contain the machine language in decimal notation (the ninth entries), from four to two. (Three will save 768 bytes; one only 256.) ■

```
10999 ' SAVE AND RESTORE SCREEN SUBROUTINE, REQU.MEMORY
      SIZE : 31699
11000 GOTO 11020  ' ENTRY POINT 1 TO SAVE SCREEN
11011 GOTO 11100  ' ENTRY POINT 2 TO RESTORE SCREEN
11020 IF ZZ$="DONE" THEN 11080
11030 DATA 33,0,60,17,254,123,1,0,4,237,176,201
11040 DATA 33,254,123,17,0,60,1,0,4,237,176,201
11050 FOR ZZ=31700 TO 31723
11060 READ Z : POKE ZZ,Z : NEXT ZZ
11070 ZZ$="DONE"
11080 POKE 16526,212 : POKE 16527,123 : Z=USR(0)
11090 RETURN
11100 POKE 16526,224 : POKE 16527,123 : Z=USR(0)
11110 RETURN   ' END OF THE SUBROUTINE
```

Listing 1.

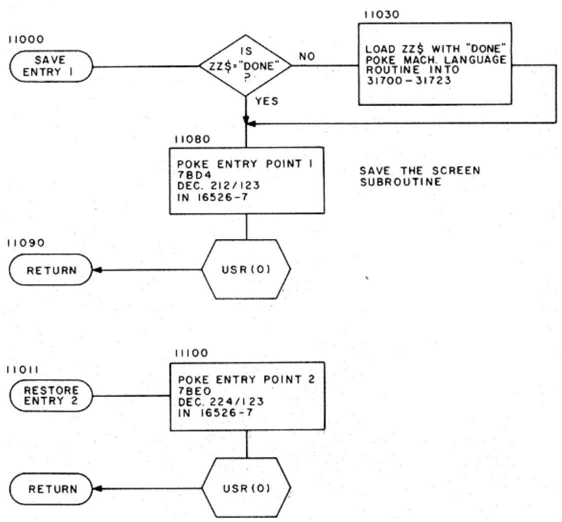


Figure 1.

```
                100  ORG   7BD4H
7BD4  21003C    110  LD    HL,3C00H  ; FROM SCREEN
7BD7  11FE7B    120  LD    DE,7BFEH  ; TO SAVE AREA
7BDA  010004    130  LD    BC,1024   ; 1024 BYTES
7BDD  EDBO      140  LDIR            ; BLOCK MOVE
7BDF  C9        150  RET             ; GO BACK
7BEO  21FE7B    160  LD    HL,7BFEH  ; FROM SAVE AREA
7BE3  11003C    170  LD    DE,3C00H  ; TO SCREEN
7BE6  010004    180  LD    BC,1024   ; 1024 BYTES
7BE9  EDBO      190  LDIR            ; BLOCK MOVE
7BEB  C9        200  RET             ; GO BACK
                210  END
```

Listing 2.

```
100 CLEAR 200:CLS
110 FOR Y=0 TO 46 STEP 2 : FOR X=0 TO 126 STEP 2 : SET(
    X,Y) : NEXT X,Y
120 FOR Y=47 TO 0 STEP -2 : FOR X=127 TO 0 STEP -2 :SET
    (X,Y):NEXT X,Y
130 GOSUB 11000
140 PRINT@522,"     WE HAVE SAVED THIS PATTERN     ";
150 FOR T=1 TO 1000:NEXT T
160 CLS:PRINT@514,"     PRESS 'SPACE' TO RESTORE THE SC
    REEN";
170 A$=INKEY$:IF A$ = "" THEN 170
180 GOSUB 11011 : GOTO 150
```

Listing 3.

80 microcomputing™ bookshelf

Z80 BOOKS

NEW!

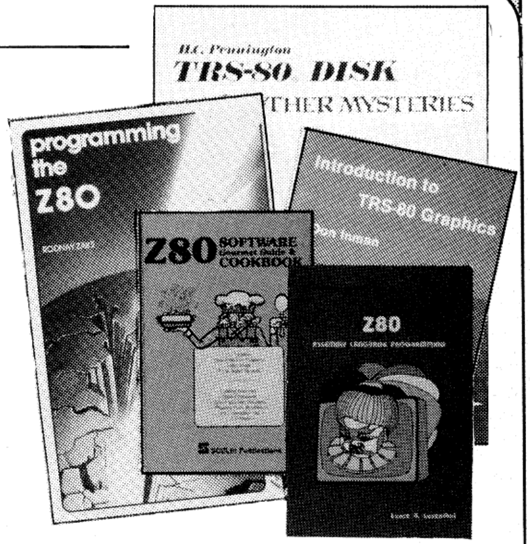
● **MICROSOFT BASIC DECODED AND OTHER MYSTERIES**—BK1186—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES*! Contains more than 6500 lines of comments for the disassembled Level II ROMs, six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flow charts for all major routines give the reader a real insight into how the interpreter works. $29.50 (Available after December 20th).

● **TRS-80 DISK AND OTHER MYSTERIES**—BK1181—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

● **PROGRAMMING THE Z-80**—BK1122—by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **Z-80 ASSEMBLY LANGUAGE PROGRAMMING**—BK1177—by Lance A. Leventhal. This book thoroughly covers the Z80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $16.99.*

● **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1045—by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $16.99.*



● **INTRODUCTION TO TRS-80 GRAPHICS**—BK1180—by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use. $8.95.*

BASIC & PASCAL

● **BASIC BASIC (2ND EDITION)**—BK1026—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $9.45.*

● **LEARNING LEVEL II**—BK1175—by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● **THE BASIC HANDBOOK**—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

● **INTRODUCTION TO PASCAL**—BK1189—by Rodnay Zaks. A step-by-step introduction for anyone wanting to learn the language quickly and completely. Each concept is explained simply and in a logical order. All features of the language are presented in a clear, easy-to-understand format with exercises to test the reader at the end of each chapter. It describes both standard PASCAL and UCSD PASCAL, the most widely used dialect for small computers. No computer or programming experience is necessary. $12.95.*

● **ADVANCED BASIC**—BK1000—Applications, including strings and files, coordinate geometry, area, sequences and series, simulation, graphing and games. $9.65*.

"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf ● Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All above add $1.00 handling. Please allow 4–6 weeks for delivery. Questions regarding your order? Please write Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing™ bookshelf

GAMES

● **MORE BASIC COMPUTER GAMES** – BK1182 – edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games.* Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. $7.50.*

● **WHAT TO DO AFTER YOU HIT RETURN** – BK1071 – PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

● **BASIC COMPUTER GAMES** – BK1074 – Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

SPECIAL INTERESTS

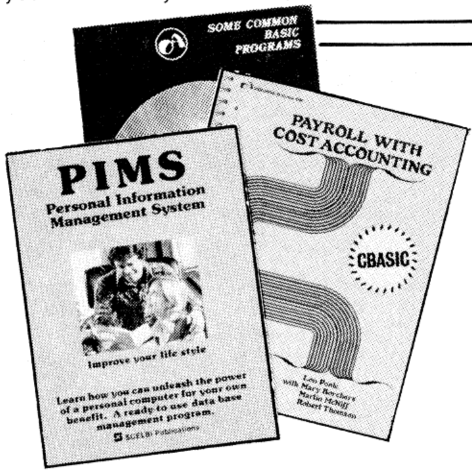
● **THE CP/M HANDBOOK (with MP/M)**—BK1187—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS. $13.95.

● **HOW TO MAKE MONEY WITH COMPUTERS** – BK1003 – In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

● **HOW TO SELL ANYTHING TO ANYBODY** – BK7306 – According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune – and how you can, too. $2.25.*

● **THE INCREDIBLE SECRET MONEY MACHINE** – BK1178 – by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. $5.95.*

BUSINESS



● **PAYROLL WITH COST ACCOUNTING – IN BASIC** – BK1001 – by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in **CBASIC**). $20.00.*

● **SOME COMMON BASIC PROGRAMS**—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $14.99 paperback.

● **PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM** – BK1009 – Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $11.95.*

"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All above add $1.00 handling. Please allow 4–6 weeks for delivery. Questions regarding your order? Please write Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

HOBBYWORLD ELECTRONICS, INC.

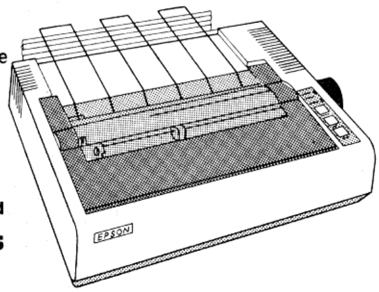
19511 BUSINESS CENTER DRIVE, DEPT. V2 • NORTHRIDGE, CALIFORNIA 91324

EPSON MX-80 Printer

A low cost printer with all the features you've been asking for!

- 9 x 9 dot matrix print head
- 40-66-80 and 132 column printing widths
- Microprocessor controlled
- 80 CPS, bidirectional, logic seeking
- Programmable forms control
- Upper and lower case, with descenders
- 64 graphics characters
- Centronics-type parallel interface, standard
- Built-in self test mode
- Wt 12 lbs

Cat No. 2886 **$579.95**



16K MEMORY ADD-ON

for the TRS-80 **$32.00**

Everything you need to upgrade your system! Includes 4 pages of illustrated instructions. Complete with RAMS and pre-programmed jumpers. No special tools required! Wt 4 oz.

Cat No.	Description
1156	For TRS-80 Keyboard Unit
1156A	For TRS-80 Exp. Interface purchased before 4/1/79
1156B	For TRS-80 Exp. Interface purchased after 4/1/79
1156C	APPLE II
1156D	EXIDY

PROGRAMMERS GUILD IRV

IRV allows the user to define keys to represent special actions or functions. Using IRV, one keystroke can represent a whole line of code or data! Define those often-used commands with IRV to increase the efficiency, action, and enjoyment of your system. IRV resides in high memory and is self-patching. Includes: auto cassette control, flashing cursor control, auto repeat and more! Wt 5 oz.

Cat No. 2776 TRS-80, L2, 16-48K, cass $24.95
Cat No. 2777 TRS-80, L2, 16-48K, disk $29.95

INTERPRO Atlantian Odyssey

Embark upon a journey that takes you through three oceans, and to the tops of towering mountains. Visit underwater cities and see strange painted shapes that will lead you to the lost city of Atlantis. Games in progress can be saved on tape and continued at a later time. Wt 4 oz.

Cat No. 2855 TRS-80, L2, 16K, cass **$14.95**

the Green Screen

for the TRS-80

If you've ever wished your TRS-80 video display could be improved, here's the inexpensive answer! The Green Screen attaches in seconds, instantly improves readability, and allows you to view the screen longer with reduced eye strain! Wt 6 oz.

Cat No. 2790 **$12.95**

SUB LOGIC Flight Simulator

You have a real-time, 3-D view of flight that's extremely realistic, including a fully equipped panel with stall warning, turn indicator, radar map, bomb and ammo indicators, and control position.

Once you gain confidence, your mission will be to fly into enemy territory, destroy the enemy fuel depot, and return safely. Defeat the enemy fighters and become a British ace! Wt 8 oz.

Cat No. 2850
TRS-80, L2 & L3, 16K cass **$24.95**

INTERPRO Ultra-Mon

A ROM independent utility for the machine language programmer. Set breakpoints, relocate or disassemble a program. Displays, traces, even provides hard copy when desired. Single step through ROM and watch each register as it alters. If you're a programmer, you need Ultra-Mon! Wt 4 oz.

Cat No. 2854 TRS-80, L2, 16K, cass **$24.95**

BASF 5¼" DISKETTES

$35 Box/10

Soft sector, double density, single sided. Use for TRS-80, Apple, Atari.

Cat No. 2746 Box of 10 diskettes.

AUTOMATED SIMULATIONS Morlock's Tower

The mad wizard Morlock has vowed to destroy the peaceful town of Hagedorn. Can you find, and destroy, the Wizard in time?

Find magical treasures and battle monstrous opponents along the way. Features three skill levels, a unique scoring system, and more than one way to win! Wt 5 oz.

Cat No. 2832 TRS-80, L2, 32K + TRSDOS $19.95
Cat No. 2833 TRS-80, L2, 16K cass $14.95

HOW TO ORDER

Minimum Order $15.00. Order toll-free by phone or by mail, or at our retail stores. Pay by check, Mastercard, Visa or C.O.D. Please include expiration date with charge-card orders. U.S. dollars only. Include phone # and magazine issue you are ordering from. Add $1.25 for C.O.D. and shipping charges from rates below.

Shipping Rates: USA
Ground: $2.25 for first 2 lbs. and 40¢ each addt'l lb.
Air: $3.25 for first 2 lbs. and 70¢ each addt'l lb.

Shipping Rates: Foreign
Ground: $3.00 first 2 lbs. and 60¢ each addt'l lb.
Air: $11.25 first 2 lbs. and $5.00 each addt'l lb.

Prices valid thru month of magazine issue. Some items subject to prior sale or quantity limits. HobbyWorld is not responsible for typographical errors. 120 Day Guaranteed Satisfaction. Exception: Partially assembled kits, abuse or misuse.

1981 will be an exciting year for computer enthusiasts. Many terrific new products will be introduced to make your personal computing more enjoyable.

Hobbyworld is entering its fourth year of providing quality products and great service to our customers nationwide. To reaffirm our commitment to state-of-the-art computer electronics, we have adopted the new logo you see at the top of this ad. But that's not all...

Our latest catalogue contains dozens of new products for your TRS-80 owners, and some new low prices on other items. There's even a new section, the "Bargain Basement", with some really great buys for you bargain hunters.

So if you haven't received your free Hobbyworld catalogue yet, fill in the coupon below, phone, or circle our reader service number TODAY!

Utilities, Business, Sci-fi, Games, Education, and much more software for the TRS-80 is available at HobbyWorld!

FREE HOBBYWORLD CATALOG

Name ____________________
Company ____________________
Address ____________________
City ____________________
State __________ Zip __________

Call Toll-Free:
USA (800) 423-5387

Local & Outside USA:
(213) 886-9200

$uper $avings on printers & disks for the TRS-80™

CENTRONICS 779

Same as Radio Shack Line Printer I
LIST PRICE $1350 **OUR PRICE $849**
(Ship freight collect)

CENTRONICS 737

Featuring Correspondence, Quality and Proportional Spacing
LIST $995 **OUR PRICE $829**

CENTRONICS 730

Same as Radio Shack Line Printer II
LIST $795
OUR PRICE $649
(add $7.50 for shipping)

CENTRONICS 704-11
(same as 703-9)
CENTRONICS PARALLEL INTERFACE
Ideal for TRS-80

180 CPS Logic-Seeking Tractors
Adjustable to 16"
Former List Price $2975

NOW ONLY $1695

DISK DRIVES FOR TRS-80
5 1/4" Disk Drives
Ideal for TRS-80

Featuring MPI-51 Drive, 40 Track Capability, Fast Seeking

$499 VALUE

MMM PRICE ONLY $329
2 for $638

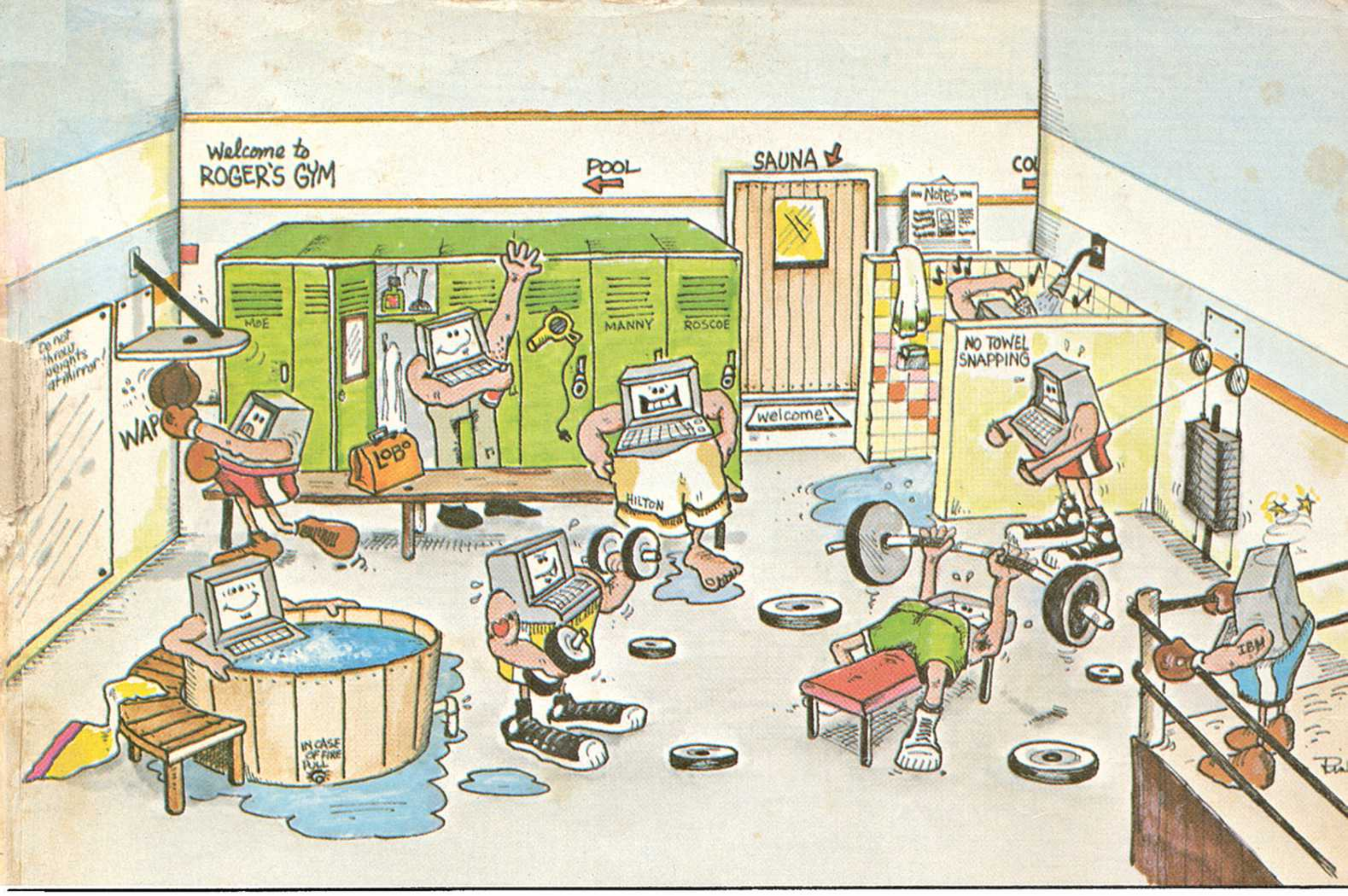
✓24

TM TRS-80 is a trademark of Radio Shack

MiniMicroMart, Inc. 1618 James St., Syracuse, NY 13203. (315) 422-4467 TWX 710 541-0431

LIST OF ADVERTISERS

***This advertiser prefers to be contacted directly.**



LOBO'S NEW LDOS™ Puts muscle in your TRS-80*

LOBO DRIVES' new LDOS™ Disk Operating System is loaded with outstanding features that will enable you to realize the full power and potential of your TRS-80.* With LDOS, you can support up to eight drives (5¼ and 8-inch drives, double-sided drives, double-density drives, 80-track drives), including the new 8-inch and 5¼-inch Winchester fixed disk drives, in any combination.

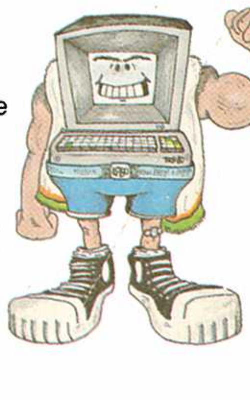
Other LDOS muscle building features include: ISAM accessing techniques; keyboard typeahead; Graphic string packer; Dated files, Marked files; File transfer by class; Built-in lower case display drivers; Non-breakable AUTO and DO commands, and many, many more.

LDOS is the perfect operating system to use with your LOBO DRIVES LX-50 or LX-80 expansion interface and disk drive subsystems. There's even an 800 number for instant service. To find out how you can put more muscle into your TRS-80, contact your nearest LOBO Drives dealer or call or write.

LDOS is available:

A. Operating System Diskette with Master Reference Manual $139

B. Master Reference Manual Only $25



* TRS-80 is a registered trademark of Radio Shack, a Tandy Company.

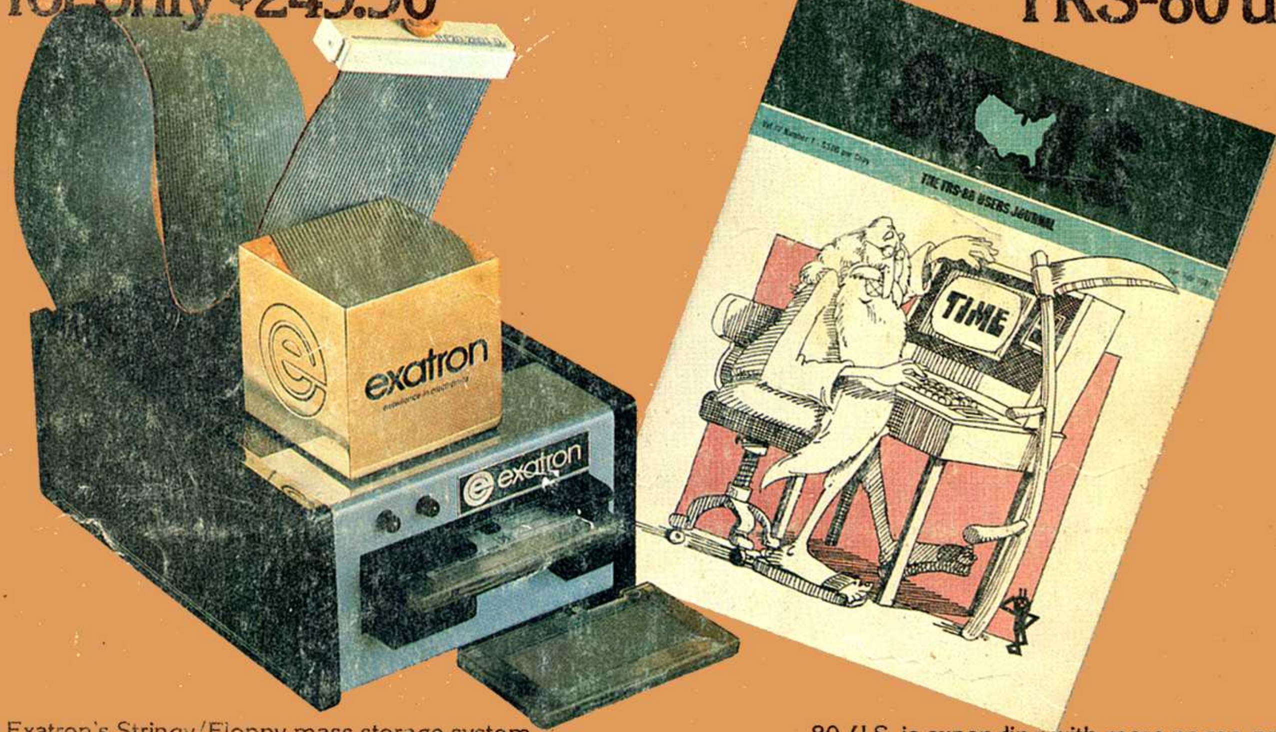
LOBO drives INTERNATIONAL

LOBO DRIVES, INT'L ✓15
354 South Fairview Ave.
Goleta, CA 93117
(805) 683-1576

Two proven ways to expand your TRS-80* capability

Exatron's Stringy/Floppy... speed, capacity and reliability for only $249.50

80-U.S. TRS-80 User's Journal New "How-To" info for the serious TRS-80 user



Exatron's Stringy/Floppy mass storage system gives you the speed, capacity and reliability of a mini-disk system at far less cost. Thousands of ES/F users agree.

Here's your complete Starter System:

- ES/F Operating Manual
- Basic ES/F System ($249.50)
- 10 Blank Wafers
- ES/F Monitor Program
- Tutorial Demo Program
- 2 for 1 Bus Connector
- Data I/O Program
- FREE 1 Year Subscription to 80-U.S., the User's Journal
- Complete Info Package
- Complete Starter Kit: $299.50

CALL OUR FREE HOTLINE
(800) 538-8559
IN CALIFORNIA: (408) 737-7111

Or contact:

exatron ✓94
181 Commercial Street
Sunnyvale, CA 94086

TRS-80 Trademark of Tandy Co.

80-U.S. is expanding with more pages, programs and color with heavy emphasis on advanced TRS-80 applications... plus a regular Exatron Stringy/Floppy feature.

Special Introductory offer $9.95 per year

We'll send you six issues of 80-U.S. for only $9.95... almost half the regular price. If you purchase an Exatron Stringy/Floppy System, we'll send them FREE.

OFFER VALID UNTIL MARCH 31, 1981.

Send to:

80-U.S.

80-U.S. Journal ✓33
3838 South Warner Street
Tacoma, WA 98409
(206) 475-2219

Sign me up! I can't resist at $9.95!

☐ Check ☐ Money Order ☐ M/C or Visa

CARD NUMBER ______ EXP DATE ______

NAME ______

ADDRESS ______

CITY ______ STATE ______ ZIP ______

Offer good only in the United States.